commodore Magazine

Año I - Número 1 - 250 Ptas.



commodore Magazine

Sumario

Commodore Magazine es una publicación de Ediciones y Suscripciones, S. A. C/Jerez, 3. Madrid-16.
Presidente:
Fernando Bolín.
Director Editorial:
Norberto Gallego.
Director:
Alejandro Diges.
Redacción:
(91) 250 15 92.
Aníbal Pardo.
Gumersindo García.
Roberto Menéndez.
Simeón Cruz.
Miguel Angel de Frutos.
Manuel Arias.
Diseño:
R. Segura.
Circulación:
Luis Carrero. 4572617
Suscripciones: (91) 250 15 93.
Antonio Zurdo.
Producción:
Miguel Onieva.
Publicidad Madrid: Jerez, 3. Madrid-16.
María José Martín. (91) 457 45 66.
Nieves Fernández. (91) 457 45 66.
Publicidad Barcelona: Tallers, 62-64. Barcelona-1.
Enrique Alier. (93) 302 36 48.
Distribuye: SGEL. Avda. Valdelaparra, s/n. Alcobendas, Madrid.
Imprime: Novograph, S.A. Ctra. de Irún, Km. 12,450. Madrid.
Fotomecánica: Karmat. Pantoja, 10. Madrid.
Depósito Legal: M - 6622 - 1984

Año 1
Numero 1

Esta revista no mantiene relación de dependencia de ningún tipo con respecto de los fabricantes de ordenadores Commodore Business Machines ni de sus representantes.

Editorial

Salimos a la calle con vocación de convertirnos en la revista de cabecera de todos los usuarios de ordenadores Commodore *Nuestro objetivo lo tenemos claro: ser una publicación útil y amena, un canal de comunicación entre sus lectores.*

Commodore Magazine *tiene desde su nacimiento dos rasgos que nos parece importante destacar. El primero es el hecho de estar elaborada totalmente en España, por un equipo humano nacional. No es la traducción ni adaptación de ninguna publicación extranjera. Sin embargo, no vamos a descartar artículos puntuales de colaboradores extranjeros que, por su interés, merezcan aparecer en estas páginas. La segunda seña de identidad de* Commodore Magazine *será su absoluta independencia respecto de Commodore Business Machines o de sus representantes. Lo cual no quita que aspiremos a mantener con ellos una cordial relación.*

Deseamos ofreceros vuestra revista. Pretendemos que Commodore Magazine *sea una ventana abierta a todos cuantos utilizamos ordenadores* Commodore. *Esperamos la colaboración de los lectores porque de ella dependerá la vitalidad de la revista. Y deseamos que esta invitación no caiga en saco roto. La intención de los concursos que iniciamos en este número, y a los que seguirán otros, no es otra que la de ser un aliciente más para estimular el esfuerzo de los lectores. Además, a nadie amarga un dulce.*

Traeremos a las páginas de Commodore Magazine *muchos programas y, cómo no, revisiones del* ***software, hardware,*** *dispositivos y todo lo que veamos en el mercado nacional. Ocasionalmente, nos ocuparemos de lo que ocurre fuera de nuestro país, pero dentro de unos límites marcados por su interés real. Poco nos importan los fabulosos productos que uno puede comprar en Estados Unidos si, por una serie de razones, resulta improbable que se comercialicen en España. Bastará, pues, con conocer su existencia y reseñarla para azuzar un poco al mercado.*

Este primer número se ha centrado, principalmente, en la gama baja de ordenadores Commodore, *concretamente en los modelos* VIC-20 *y* CBM 64 *pero no es nuestra intención relegar a los demás modelos de la marca, algunos de ellos con una buena base instalada en España.*

Entendemos que no todos los lectores tienen el mismo nivel de conocimientos, y que los gustos y preferencias son también múltiples. Por eso, intentaremos que el contenido editorial sea lo más variado posible, para contentar a unos y a otros.

Por descontado queda que nuestros lectores deberán disculpar, al menos en este primer número, algunos pequeños defectillos que procuraremos ir corrigiendo en lo sucesivo. Ahí está el desafío que hoy asumimos. Animo, entre vuestra colaboración y nuestro trabajo, vamos a lograr la mejor revista de habla hispana para los usuarios de Commodore.

Claves para introducir en el ordenador los programas que aparecen en la revista.

COMO SE VE	COMO SE TECLEA	EFECTO CONSEGUIDO
	SHIFT + CLR	(LIMPIA PANTALLA)
	CLR	(HOME)
	CRSR	(CURSOR ABAJO)
	SHIFT + CRSR	(CURSOR ARRIBA)
	CRSR	(CURSOR DERECHA)
	SHIFT + CRSR	(CURSOR IZQUIERDA)
	CTRL + 1	(NEGRO)
	CTRL + 2	(BLANCO)
	CTRL + 3	(ROJO)
	CTRL + 4	(CIAN)
	CTRL + 5	(VIOLETA)
	CTRL + 6	(VERDE)
	CTRL + 7	(AZUL)
	CTRL + 8	(AMARILLO)
	CTRL + 9	(CARACTER INVERSO)
	CTRL + 0	(CARACTER NORMAL)

Existen en la memoria de un ordenador determinadas posiciones específicas, que controlan algunas de las características de la máquina. Si se conocen cuáles son, qué característica de la máquina determinan y cómo actuar sobre ellas, se puede entonces alterar esa característica a voluntad.

Conocer estas posiciones clave va a permitir al programador sacar el máximo partido de las posibilidades del ordenador y puede serle de gran utilidad en muchos programas.

Un ejemplo para el **VIC 20**, son las posiciones de memoria 36897 y 36896, que controlan, respectivamente, las posiciones vertical y horizontal del recuadro de trabajo sobre la pantalla del televisor.

Variando el contenido numérico de estas posiciones, mediante "POKES", se varía la posición del recuadro.

A continuación damos un sencillo programa que permite situar el recuadro en cualquier posición, dentro de la pantalla, actuando sobre cuatro teclas del teclado.

Como, en todo momento, el programa le dice qué valor hay metido en las posiciones de memoria 36897 y 36896, podrá apuntar esos valores cuando la posición del recuadro le interese.

VIC - 20

```
0 REM***POSICION PANTALLA***
5 MU=PEEK(36897):MA=PEEK(36896)
10 PRINT"⌐U=ARRIBA":PRINT"D=ABAJO":PRINT"R=DERECHA":PRINT"L=IZQUIERDA"
15 PRINT"SQQQQQQQ▶▶▶▶PEEK(36897)="MU:PRINT"▶▶▶▶PEEK(36896)="MA
20 GETA$:IFA$=""THEN20
30 IFA$="U" THEN MU=MU-1
40 IFA$="D" THEN MU=MU+1
50 IFA$="L" THEN MA=MA-1
60 IFA$="R" THEN MA=MA+1
80 IF MU<0 OR MU>125 THEN 20
90 IF MA<0 OR MA>50 THEN 20
100 POKE36897,MU:POKE36896,MA
120 GOTO15
```

Algunas veces, cuando se busca una agradable presentación de un programa en la pantalla del televisor, se hecha de menos el no disponer de varios tamaños de caracteres, para construir títulos o encabezamientos. Este programa puede ser útil en estos casos, pues permite obtener caracteres de doble altura. La construcción de estos caracteres de doble altura se lleva a cabo poniendo, uno encima de otro, dos caracteres de altura normal, y luego, mediante un POKE, modificando el código de caracteres en pantalla, con lo que los dos caracteres de altura normal se funden en un solo carácter de doble altura. El programa incluye una rutina de demostración, a partir de la instrucción 210, en la que se muestra la correspondencia entre caracteres de doble altura y parejas de caracteres estandar. Una vez pasado el programa a la memoria del ordenador, escriba RUN. Aparecerán en la pantalla dos líneas de caracteres estándar y luego, se fundirán en una línea de caracteres de doble altura. Para volver a los caracteres estándar pulse 1, y para ver los de doble altura, pulse 2. Cuando se haya familiarizado con la tabla de correspondencias, podrá borrar la rutina de demostración (líneas 210 y siguientes) y pasar sólo la rutina para crear los caracteres. Una vez pasada esta rutina escriba sus títulos que serán dos líneas de caracteres normales, una encima de otra, y escribiendo POKE 53272,(PEEK(53272)AND240)+12 tendrá caracteres de doble altura. Para volver al formato normal utilice RUN/STOP,RESTORE o bien POKE 53272,21. Si cambia el valor 53248 en las líneas 50 y 60, podrá acceder a otros caracteres de doble altura. Si lo sustituye por 55296 tendrá minúsculas y si utiliza el valor 53760 tendrá gráficos de doble altura.

```
5 REM***CARACTERES DE DOBLE ALTURA***
10 POKE52,48:POKE56,48:CLR
20 POKE56334,PEEK(56334)AND254
30 POKE1,PEEK(1)AND251
40 FORG=0TO255
50 POKE12288+G*2,PEEK(53248+G)
60 POKE12289+G*2,PEEK(53248+G)
70 NEXTG
80 POKE1,PEEK(1) OR 4
90 POKE 56334,PEEK(56334) OR 1
200 REM
210 REM***RUTINA DEMOSTRACION***
220 REM
225 PRINT"⌐"
230 FOR G=0 TO 31
240 POKE 1024+G,G*2:POKE 55296+G,1
250 POKE 1064+G,G*2+1:POKE 55336+G,1
260 NEXTG
270 POKE 53272,(PEEK(53272)AND240)+12
280 GET A$:IF A$=""THEN280
300 IF A$="1" THEN POKE53272,21
310 IF A$="2" THEN POKE53272,(PEEK(53272)
   AND240)+12
320 GOTO280
```

COMMODORE 64

NUEVOS MODELOS



Fue uno de los primeros fabricantes de microordenadores que tomaron el mercado europeo con seriedad. Mientras otras marcas estaban preocupadas por dominar y vender en el mercado norteamericano, **Commodore International** fue asentando su imperio a este lado del océano. Decidido, posteriormente, a competir también en aquel mercado, logró impactar con sus **VIC-20** y **Commodore 64.** Pero la competencia es dura y son muchos los que tropiezan en el camino. Después de las notorias pérdidas acumuladas por otros fabricantes de ordenadores personales, lease **Texas Instruments, Atari** o **Mattel, Commodore** toma la iniciativa de presentar nuevos y revolucionarios productos a principios del pasado mes de enero. No obstante, se afirma que durante el año 83 han vendido un millón doscientas mil unidades **64**. Las nuevas máquinas responden a las denominaciones **264** y **364.**

Aunque no se sabe certeramente cuando comenzarán los primeros envíos, se rumorea el próximo abril como fecha de llegada al mercado. De todas formas, portavoces de la compañía han afirmado que no tienen demasiada prisa *"dada a alta demanda actual del modelo* **64**, *superior a su capacidad de fabricación*", añadiendo que *"esperamos que con el* **64** *y el* **264** *vamos a dominar el mercado del ordenador doméstico durante 1984"*. Esta afirmación la hace **Don Richard**, presidente de **Commodore**. La nueva línea **264** está concebida como una adición de la serie previamente formada por **VIC-20** y **64**. Entre otras cosas aporta *software* incorporado en su interior, de modo opcional. En realidad los diferentes modelos **264** diferiran en este *soft*, pero de momento no han sido desvelados más detalles referentes a él, sólo que se orientarán en gran medida a mejorar la productividad personal. Una cifra ha salido a relucir, para aquel entonces, la división **Commodore Software** tendrá disponibles unos 30 productos incluidos en cartucho cinta o disco. La clara intención de la compañía es ofrecer ordenadores personalizables por medio del *software.* Una aclaración interesante es la que asegura la disponibilidad de cartuchos para el **64**, que contienen el mismo *software* destinado al **264.**

Las características técnicas más destacables de los **264** incluyen 64 *Kbytes* de RAM, de los cuales 60 quedan libres para la programación en BASIC, la capacidad para que la pantalla trabaje de un modo similar a una ventana, 8 teclas de función reprogramables, tecla de ayuda (HELP). Aparecen 4 teclas dedicadas al movimiento del cursor en pantalla, al contrario de lo que sucede con los **VIC-20** y **64**, que sólo llevan 2.

La compatibilidad con los periféricos ya existentes es casi total. El color no ha sido olvidado, cada vez toma un mayor protagonismo en esta gama de ordenadores. La paleta ofrecida por el **264** es amplia en posibilidades, un total de 128 colores, desglosados en 16 primarios y 8 niveles de luminancia. La alta resolución gráfica y la posibilidad de compartir texto y gráficos, al mismo tiempo, es otra realidad; hablándose de un total de 200 por 320 puntos en pantalla.

El sonido está igualmente asegurado, mediante la incorporación de 2 generadores de tono.

Para completar su manejo, el BASIC dispondrá de unos 75 comandos y además otros 12 del monitor de lenguaje máquina incluido. El BASIC incluido la es la versión 3.5.

El **CV364**, teniendo también 64 *Kbytes* de memoria, incorpora un teclado numérico separado del cuerpo principal de teclas. El microprocesador utilizado es el mismo del **C264**, fabricado en tecnología HMOS, con denominación 7501. Este modelo posee capacidad interna para la síntesis de voz, con un vocabulario de unas 250 palabras incluidas, ampliables a partir de disco o cartuchos. Además de los *ports* clásicos, el destinado al Modem está presente, como el **264**. De igual manera, el **CV364** podrá compartir prácticamente todos los periféricos de la firma existente.

MODULO SINTETIZADOR DE VOZ

Una novedad, que suponemos llegará a ser muy apreciada por determinado círculo de usuarios, es el

Noticias y Novedades

módulo productor de síntesis de voz para el **CBM 64.** Su utilización parece sencilla en principio, basta con enchufarlo directamente al *port* del usuario. A su vez, dispone de un *port* adicional, destinado al alojamiento de nuevos cartuchos, sean "mudos" o "parlantes".

Según afirma el fabricante, su vocabulario inicial queda compuesto por 235 palabras, pronunciadas por una voz femenina extremadamente agradable, aunque, eso sí, suponemos que en inglés. Igualmente se puede elegir que la voz se genere a diferentes velocidades, a saber: lenta, normal o rápida. Otras palabras pueden ser programadas directamente desde el BASIC, ensamblador o ambos. Las posibilidades se multiplican, uno podrá desarrollar programas que contengan música, gráficos y una adorable voz que nos acaricie los tímpanos con su cálido verbo. ¿Quién da más?

Pero las posibilidades no acaban aquí. Mediante la salida de audio podemos obtener el sonido a travé de nuestro equipo Hi-fi, o sino por el televisor.

El fabricante ha prometido que otras voces diferentes están por llegar, entre ellas, la voz masculina, personajes del cómic, etc., en forma de los discos y cartuchos mencionados. De cara a los programas educativos, está previsto disponer del alfabeto, números, animales y posibilidad de deletreo.

Los cartuchos que puede soportar el módulo de voz, podrían almacenar hasta 128 *Kbytes* de información.

LOGO PARA COMMODORE

Se trata de uno de los lenguajes de programación que mayor auge está adquiriendo en los últimos tiempos. Su inmediata aplicación en fines educativos le ha hecho notorio, desde que fuera desarrollado en el **Instituto Tecnológico de Massachusets** por **Seymur Papert.** En esta ocasión es **Commodore** quien dispone de su propia versión, destinada a los **64** y **264**. De todas maneras, no es el primero en disponer de él para ordenadores de esta gama, por ejemplo Atari lo tuvo hace años para su malogrado **TI-99/4.**

Commodore destaca, como una de sus principales características, la disponibilidad de 170 *Kbytes* en *diskette.*

Por el momento, avancemos algunas de las características, con la promesa de que muy pronto haremos una más amplia descripción en estas páginas, cuando el producto esté disponible en nuestro país. Volviendo sobre los rasgos destacables del **Commodore LOGO** diospone de: 7 duendes programables, capacídades musicales, conservación de dibujos en *diskette, interface* de lenguaje ensamblador, aritmética en coma flotante, seguimiento del programa paso a paso y posibilidad de incluir comentarios.

NUEVO CURSO DE BASIC

Lo que sí está ya disponible en nuestro mercado es el curso de introducción al BASIC, en su primera parte, con destino al **CBM 64.** Al contrario de lo que ocurre con el curso para el **VIC-20**, en el caso presente no se incluyen las cintas con programas, sino que han sido incluidos los listados, para mayor gozo del aprendiz. El P.V.P. informado es de 3.000 pesetas.

MAGIC DESK

El pupitre mágico es el nombre que recibe el nuevo *software* de **Commodore,** destinado a convertir al **64** en un duro rival del **Makintosh** de **Apple Computer.** Se trata de un conjunto de aplicaciones integradas, que pueden ser manejadas de un modo totalmente iconográfico, es decir, mediante la aparición de figuras alusivas a las opciones posibles en pantalla.

Al principio, se visualiza un pupitre, con sus cajones, papelera, fichero, máquina de escribir, teléfono, calculadora, reloj, libro de cuentas y la puerta de salida a la calle.

El control se realiza moviendo, a voluntad, una mano con un dedo índice que señala.

Según lo indicado por el apéndice de la extremidad superior, podremos escribir una carta o documento con la máquina de escribir. Posteriormente se puede imprimir o guardar en el fichero, para ulterior utilización. También se podría haber tirado a la papelera. Efectuar cuentas es tarea de la calculadora, o decir la hora es del reloj.

En definitiva, el **Magic Desk** permite mecanizar muchos de los trabajos rutinarios de oficina, para satisfacción de los usuarios del **64**. Más adelante ampliaremos la información, cuando el producto llegue a nuestros mercados.

MACRO ENSAMBLADOR PARA EL 64

También está disponible el Macro Ensamblador para el **64**, con nombre de referencia **ASM 6440,** soportado en *diskette.*

Aparte del compilador *Assembler 64,* acompaña un conjunto de otros programas destinados a facilitar la edición, documentación y depuración de los programas en código máquina. Su P.V.P. 7.500 pesetas.

EASY CALC RESULT

La última novedad, por este mes, se conoce bajo el nombre de **Easy Calc Result.** Se trata de una versión simplificada del **Calc Result**, que analizamos con todo detalle en la presente edición y viene únicamente en forma de cartucho enchufable al **64.**

La hoja de trabajo disponible posee una dimensión de 63 por 254 celdillas, pudiéndose almacenar su contenido en cinta o disco. El P.V.P. asignado es de 20.000 pesetas.

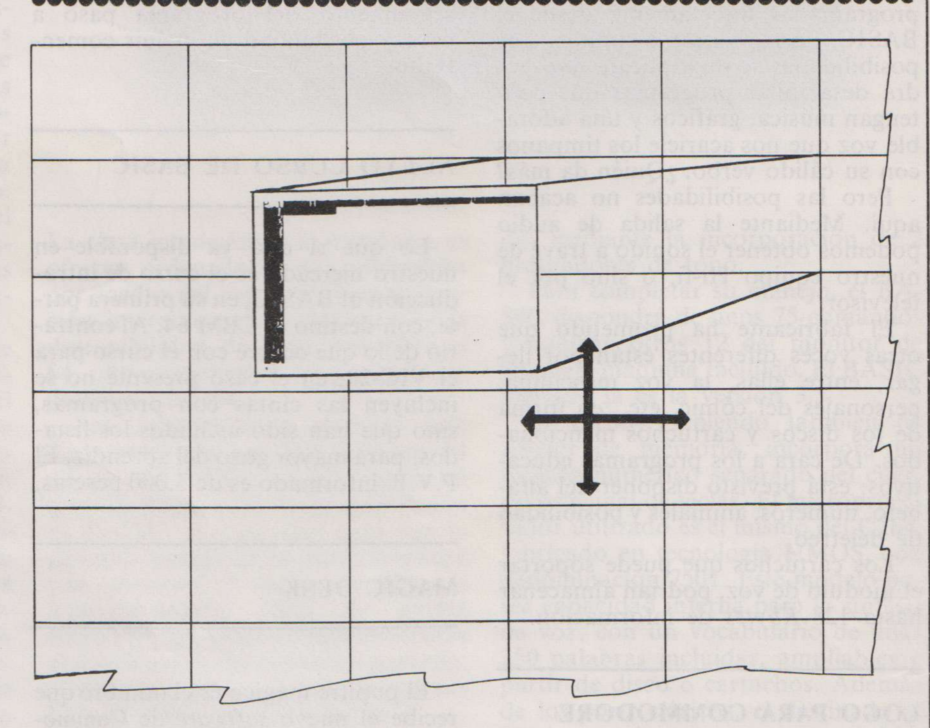
El Calc Result, desvelado

Durante el curso académico de 1978, el esfudiante **Daniel Bricklin** del **Harvard Business School** se enfrentaba regularmente a una serie de problemas, generalmente relacionados con Contabilidad, Finanzas o materias afines, que se resolvían a base de datos escritos en grandes hojas o borradores de trabajo, donde unos datos estaban relacionados con otros a través de fórmulas. Cuando algunos de los datos habían de variarse, por mala estimación inicial o por otras razones, la cantidad de trabajo que esto suponía para rehacer la hoja, a base de lápiz y calculadora, llegaba a ser agobiante.

Los ejercicios de planificación financiera, eran casos típicos de cálculo y más cálculo sobre una serie de datos básicos. El profesor quería saber "qué pasaría si..." se variaban tales o cuales previsiones o se registraban determinadas incidencias en el período que se estaba estudiando. "A veces", comenaba **Bricklin**, "sólo después de horas de trabajo advertía que centenares de resultados estaban mal por un solo error con la calculadora cometido al principio".

Aquel invierno, **Bricklin** y su colega **Bob Frankston,** ambos forofos de los micros desde muy jóvenes, trabajaron día y noche en un programa capaz de solucionar problemas similares sobre sus pequeños equipos. El resultado fue **VisiCalc** (Visible Calculator), hoy en día uno de los paquetes de software más populares para ordenadores personales. La compañía que inicialmente lo comercializó, **Personal Software Inc.**, ha cambiado su nombre por el de **VisiCorp**, y es una de las primeras firmas americanas de software. **Bricklin** y **Frankston**, hoy también propietarios de una empresa de *soft*, han pasado a engrosar la lista de hombres famosos, muy dentro de la tradición americana, que consiguen traducir en dólares (muchos) una idea feliz.

VisiCalc ha tenido desde entonces numerosas imitaciones y revisiones que han cristalizado en programas como **Multiplan, Supercalc, Logicalc**, etcétera, que, con alguna variante para aprovechar las características de determinados equipos, tienen una filosofía muy próxima a **VisiCalc**. Este



tipo de programas, conocidos en USA como *"Spreadsheets"*, se han llamado aquí hojas eletrónicas de datos, borradores electrónicos, cuadros de cálculo y otra serie de cosas. Nos quedaremos con "Cuadro de Cálculo" por ser la denominación más gráfica y obviar lo de "electrónico", que resulta un poco cargante; (ya no se oye tampoco, apenas, calificar a los ordenadores de electrónicos).

Cuestiones terminológicas aparte, lo que sí es un hecho evidente es que la aparición de este tipo de programas, ha supuesto un importante factor de impulso a la difusión cada vez más firme de los microordenadores y ordenadores personales en toda clase de ambientes. Junto al tratamiento de textos, el cuadro de cálculo se ha convertido en la aplicación por excelencia de los pequeños equipos, dentro de ese tipo de programas que podríamos llamar "serios" y que no requieren conocimiento alguno de lenguajes de programación.

Declaraciones de impuestos, planes de ventas, balances y ratios financie-

ras, carteras de valores, e infinidad de otras situaciones de gestión, tanto doméstica como de empresa, se adaptan a pedir de boca a los cuadros de cálculo. Y sobre todo, lo que les da una gran potencia y versatilidad (que tanto ha supuesto para la reputación del ordenador personal como herramienta útil), es su facilidad para examinar sin esfuerzo diversas alternativas a un mismo problema y hacer predicciones sobre la base de ciertas hipótesis traducibles en números. ¿Qué pasaría si los costes de transporte suben un 10 por 100 a partir de marzo?, ¿qué repercusión puede tener sobre el beneficio un pequeño descuento de promoción?, ¿cómo se modificará mi cifra de impuesto sobre la renta personal con la prevista subida del 6,5 por 100 en mi salario?, ¿qué inversión en valores desgravables tendré que hacer para equilibrar mi declaración?

El cuando de cálculo no pretende sustituir a esos programas que hacen listados muy bonitos y que utilizan a diario los administrativos para mantener al día los datos de la empresa. Por el contrario, se trata de una herramienta en manos de directivos, de responsables de departamentos o de empresarios y profesionales autónomos, para ayudarles en la toma de decisiones, a la hora de preparar presupuestos o de planificar la utilización de recursos.

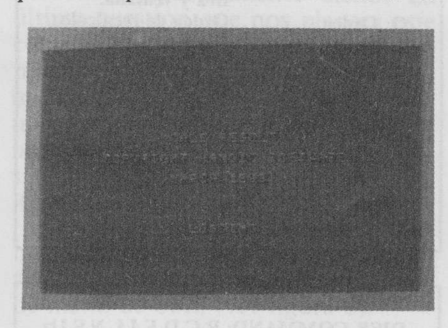
Calc Result

Calc Result, el programa de cuadro de cálculo para el **Commodore 64**, se ha desarrollado en Suecia por la compañía **Handic Software**. Escrito en **Forth**, está hecho a imagen y semejanza de **VisiCalc**. El parecido es tan notable, que parece ser que **Visicorp** piensa iniciar en Norteamérica acciones legales contra sus promotores.Pese atodo, este programa reúne algunas características de comodidad de programa y eficacia muy interesantes. Así por ejemplo, **Calc Result** hace un uso muy económico de la memoria, pues sólo utiliza espacio para aquellas posiciones realmente ocupadas por valores o fórmulas, mientras que **VisiCalc** asigna memoria de acuerdo con el tamaño físico del cuadro. Algunas otras características de este sistema son:

— Visualiza todas las opciones que ofrece cada comando, pudiéndose pedir una breve explicación de cada opción.

— Sus funciones de edición permiten cambiar, insertar o borrar textos, fórmulas o valores. La información presente en el cuadro se reorganiza cuando se introducen nuevas filas y columnas.

— Una fórmula matemática se puede reproducir cuantas veces se quiera. Se pueden manipular filas, columnas o matrices completas.



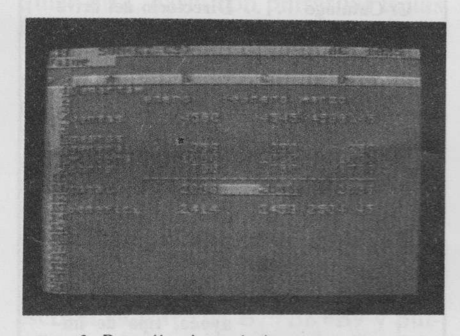


Foto 2. Pantalla de trabajo.

— La evaluación de expresiones sigue las reglas normales de aritmética (en otros programas de este tipo la evaluación sigue otros criterios).

— Permite representar los valores como histogramas de barras, con escalas definidas por el usuario, adaptadas a la magnitud de los valores. Estas tablas gráficas pueden después imprimirse.

El sistema **Calc Result** está soportado físicamente en un cartucho especial que se conecta en el panel trasero del **Commodore 64**, y un disco maestro del que es preciso sacar una copia para empezar a trabajar. El equipo mínimo requerido, aparte de esto, es un **Commodore 64** con una unidad de disco flexible **VIC 1451** y una impresora **VIC 1515/1525, 1520 ó 1526.**

Arrancando desde cero

El manual que acompaña la aplicación describe un detallado protocolo para comenzar a trabajar, partiendo de la situación "todo apagado". Siguiéndolo escrupulosamente no se precisa acudir a ninguna función ni comando externos a Calc Result. Una de las operaciones incluidas en el protocolo exige completar un "Registro de Usuario" o esquema de personalización del programa de acuerdo con la configuración que se posee. Las peticiones de datos que se efectúan son:

— Idioma (ocho posibilidades).

— Doble/Simple unidad de disco.

— Color (del fondo, del cuadro y de los ejes).

— Número de periférico de la unidad de discos.

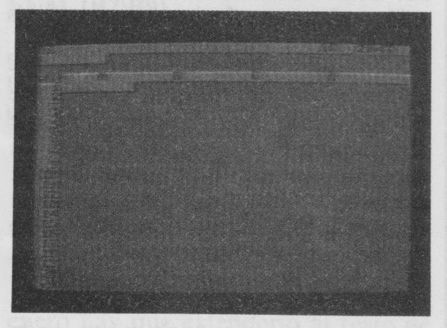


Foto 1. Pantalla sin utilizar.

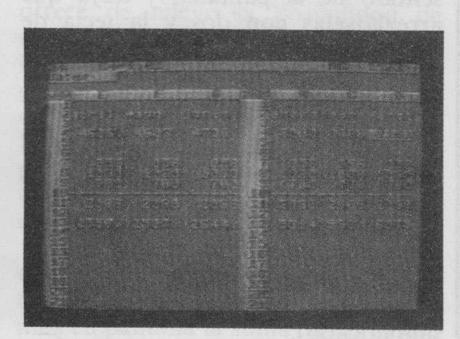


Foto 3. Pantalla compartida.

— Número de líneas del papel de la impresora.

— Número de caracteres por línea.

— Margen izquierdo.

Cubiertos todos los pasos, el ordenador exhibirá nuestra primera y prometedora pantalla de trabajo (ver foto 1).

A partir de este instante, cada una de las celdas formadas por la intersección de filas (designadas numéricamente) y columnas (designadas con las letras del alfabeto) podrán contener títulos alfanuméricos, cantidades o fórmulas que relacionen unas celdas con otras.

La pantalla sólo visualiza una pequeña parte de todo el cuadro de cálculo que el usuario tiene realmente disponible. Podemos contemplar la pantalla como si se tratara de una ventana abierta en el cuadro. Esta ventana puede desplazarse, utilizando las teclas de movimientos del cursor, tanto horizontal como verticalmente (lástima que el **Commodore 64**, no disponga de una tecla para cada sentido de la pantalla, y haya que arreglárselas con dos y la tecla de "*shift*"), para tener acceso a las 63 columnas y 254 filas que componen el área de trabajo completa (ver figura 1).

Construcción de un modelo

Vamos a tomar como ejemplo de puesta en práctica de **Calc Result**, la realización de un análisis financiero de una empresa (necesariamente muy simplificado).

Tendremos en nuestro cuadro una línea dedicada a las cifras de Ventas, un capítulo de Gastos desglosado, por ejemplo, en Gastos Generales, Mano de Obra y Otros, y una línea para los Beneficios previstos. Las relaciones iniciales de nuestro cuadro son obvias: el Total de Gastos es la suma de las cantidades correspondientes a los conceptos parciales citados, y el Beneficio, para cada uno de los meses (haremos el examen para un semestre), es la diferencia entre las Ventas y el Total de Gastos.

Inicialmente, con las teclas de direccionamiento del cursor, llevamos éste al lugar del cuadro donde deseamos empezar a escribir, por ejemplo, a A4 (es decir, a la intersección de la columna A y la fila 4), y una vez allí pulsamos la barra espaciadora y a continuación teclearemos:

Ventas

(La barra espaciadora previa le indica al **Calc Result** que se trata de un literal, no una cantidad ni una fórmula).

Al pulsar luego la tecla de cursor

SYSTEM COMMAND: B D E F G L O P Q R —

B: Blanco	**Borra el contenido de la celda donde se halla situado el cursor.**
D: Disco	**Para comunicación con el disco o registro Usario.**
E: Edición	**Para pantalla e impresora.**
F: Formato	**Celda individual.**
G: Global	**Formato global y anchura de columna.**
L: Dejar	**Título, pantalla activa y ventana.**
O: Orden	**Orden de recalculo.**
P: Página	**Funciones de página.**
Q: Salir	**Salida de programa.**
R: Recalcula	**Automático o manual.**
—:	**Repetición automática de caracteres en la celda bajo el cursor.**

DISK COMMAND: B C D E I L N S U

B: Backup	**Drive 0 al drive 1.**
C: Catálogo	**Directorio del drive 1.**
D: DIF-fichero ..	**Para grabar y cargar ficheros DIF.**
E: Borra	**Borra fichero de drive 1.**
I: Inicializa	**Drives 0 y 1.**
L: Carga	**Fichero a área de trabajo.**
N: Formatea	**Formateado de disco en drive 1.**
S: Salva	**Area de trabajo en drive 1.**
U: Registro User .	**Contiene lenguaje para pantallas de ayuda, tipo de impresora y formato de papel.**

EDIT COMMAND: C D G I M P R S T W

C: Copia	**Area de datos a otra área.**
D: Borra	**Línea o columna.**
G: Gráfico	**Histograma.**
I: Inserta	**Línea o columna.**
M: Mueve	**Area de datos a otra área.**
P: Imprime	**Hoja de trabajo o formato definido por usuario.**
R: Réplica	**Area de datos a otras áreas.**
S: Activa	**Pantalla (horizontal /vertical).**
T: Título	**Protege un título en la columna izquierda.**
W: Ventana	**Inserta ventana.**

Figura 1

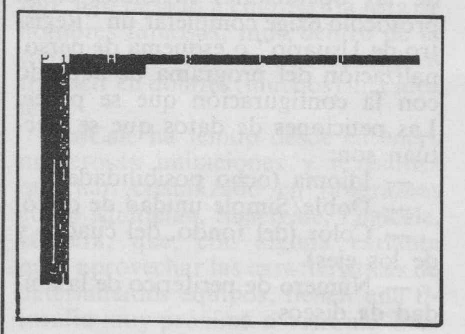


Figura 2

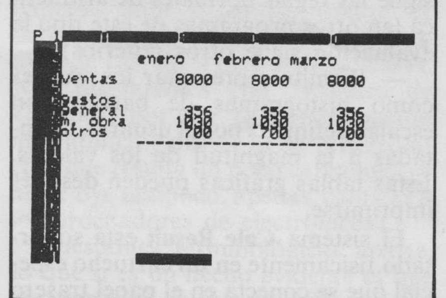
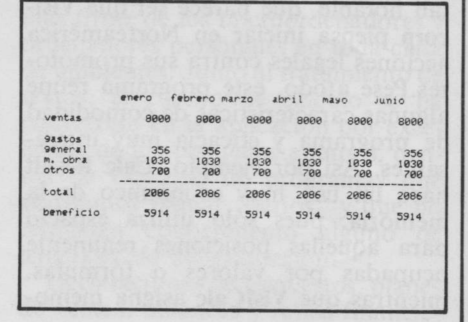


Figura 3



	enero	febrero	marzo	abril	mayo	junio
ventas	8000	8000	8000	8000	8000	8000
gastos						
general	356	356	356	356	356	356
m. obra	1030	1030	1030	1030	1030	1030
otros	700	700	700	700	700	700
total	2086	2086	2086	2086	2086	2086
beneficio	5914	5914	5914	5914	5914	5914

→ , éste se situará en B4, habiendo entrado previamente en A4 el texto "Ventas". Ahora escribimos 8.000 (esta vez sin espaciado previo), supuesto que esta es la cantidad correspondiente a enero.

Supongamos que prevemos inicialmente una cifra de ventas igual para todos los meses. Una solución evidente es ir completando el resto de los meses como hemos hecho para enero. Otra, más elegante y rápida, es utilizar el comando Replicate. Con el cursor en B4 escribiremos:

f7 E R Return Return
C4 Return G4 Return

Expliquemos un poco más estas expresiones.

f7 es la tecla del **Commodore** mediante la que se accede a cualquier comando del arsenal de que dispone **Calc Result**, y que puede verse en el recuadro. *Replicate*, como allí consta, es un subcomando de edición, de ahí que primero escribamos **"E" (edición) y luego "R"** *(Replicate)*. Aquí ya podemos observar una regla general de **Calc Result**: el acceso a cualquier comando es por la inicial de su nombre (inglés), lo que, desde luego, es un importante inconveniente de manejo del programa, por lo menos al principio.

El resto de la orden sirve para especificar la zona fuerte y la zona de destino de la copia. Los dos primeros *"Return"* especifican que el área de origen, que empieza y termina en B4 (donde está situado el cursor), mientras que la zona de destino comienza en C4 y termina en G4 (meses de febrero a junio).

Estas coordenadas con que vamos respondiendo a las peticiones del comando *Replicate*, se van reflejando en una línea de edición por encima del cuadro propiamente dicho. La última cuestión que nos plantea este comendo es si la copia ha de ser con referencias absolutas o relativas. Como en esta ocasión se trata de copiar un valor fijo (no una fórmula en que intervengan coordenadas de otras posiciones), contestaremos A, inicial de *Absolute*.

De la misma manera introduciremos, en filas sucesivas, los datos relativos a Gastos generales, de material y de mano de obra (ver figura 2).

Para obtener impresa esta imagen de la pantalla, lo mismo que cualquier otra, basta con pulsar las teclas **f7 f6**.

Como se ve, hemos añadido una raya separadora debajo de cada grupo de Gastos, y hemos introducido ya el literal Total Gastos. Ahora tenemos el cursor en B11 (la posición actual del cursor queda siempre indicada en la línea superior de la pantalla), y esa posición queremos que refleje la suma de las cantidades que figuran en B7, B8 y B9. Basta con escribir la fórmula B7+B8+B9 y pulsar a continuación *Return*. En B11 verá aparecer la cantidad de 2036 que corresponde a la suma.

Calc Result facilita mucho las cosas a la hora de introducir expresiones. Así por ejemplo, para la suma anterior ni siquiera es preciso escribir la denominación de cada celda que hay que sumar, sino que basta con ir colocando el cursor en las celdas correspondientes, y el propio programa se encarga de expresar sus coordenadas. El usuario sólo tiene que añadir los sinos aritméticos.

Otra forma de hacerlo es empleando la función incorporada SUM (ver funciones matemáticas). En este caso la fórmula se expresaría como
SUM (B7:B9)

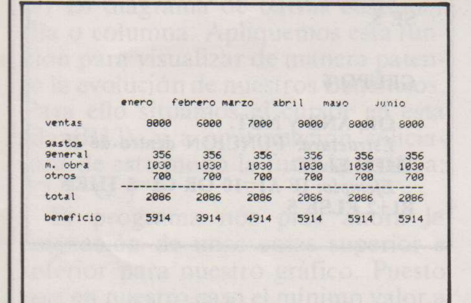
Obsérvese, pues, que es posible especificar un área de celdas consecutivas mediante la expresión "primera: última".

Reproducción de fórmulas. Referencias absolutas y relativas

El proceso seguido con el mes de enero hay que extenderlo al resto de los meses. El comando *Replicate* nos ofrece de nuevo la posibilidad de copiar la fórmula sin tener que escribirla de nuevo para cada uno de los cinco meses. Esta vez, con el cursor en B11, teclearemos:
f7 E R B11 Return Return (define la posición B11 como zona origen)
C11 Return G11 Return (zona destino, meses de febrero a junio).

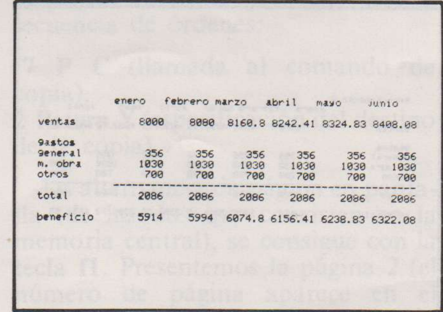
Y, a continuación, en respuesta al tipo de referencias contestaremos **R** (de *Relative*). De inmediato, tendremos en pantalla la suma de los gastos para todos los meses. El hecho de copiar una fórmula con referencias relativas significa que las coordena-

Figura 4



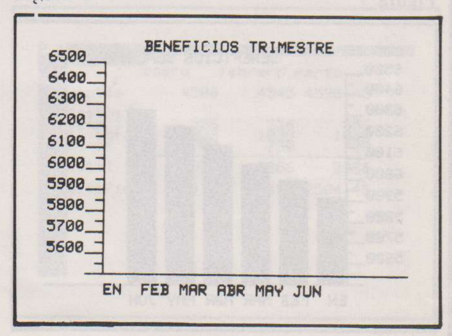
	enero	febrero	marzo	abril	mayo	junio
ventas	8000	6000	7000	8000	8000	8000
gastos						
general	356	356	356	356	356	356
m. obra	1030	1030	1030	1030	1030	1030
otros	700	700	700	700	700	700
total	2086	2086	2086	2086	2086	2086
beneficio	5914	3914	4914	5914	5914	5914

Figura 5



	enero	febrero	marzo	abril	mayo	junio
ventas	8000	8080	8160.8	8242.41	8324.83	8408.08
gastos						
general	356	356	356	356	356	356
m. obra	1030	1030	1030	1030	1030	1030
otros	700	700	700	700	700	700
total	2086	2086	2086	2086	2086	2086
beneficio	5914	5994	6074.8	6156.41	6238.83	6322.08

Figura 6



das que intervienen en ella se ajustan para cada columna. Así, la fórmula inicial para los gastos de enero, SUM (B7:B9), se transforma en febrero en SUM(C7:C9), en marzo en SUM(D7: D9), y así sucesivamente.

Añadamos ya la línea relativa a los beneficios, definiendo, en el mes de enero qué beneficio = ventas — gastos, es decir, dando a la celda B13 el valor (B4 — B11). Como ya dijimos antes, para escribir esta fórmula basta con que, cuando queremos hacer referencia a una coordenada dirijamos el cursor hasta situarlo en ella, en lugar de escribirla directamente. Esto nos evita, por una parte, manejar una notación incómoda, y por otra, posibles errores visuales.

Ahora, igual que hemos hecho con los gastos totales, reproduzcamos esta fórmula para el resto del semestre con el comando *Replicate* (y especificando igualmente que queremos referencias Relativas).

Para imprimir ahora nuestro cuadro, vamos a utilizar la orden P *(Print)* dentro de los comandos de Edición. Este procedimiento nos permite definir cuántas líneas deseamos imprimir, cuántas columnas y la anchura de cada una de éstas, con sólo responder una serie de preguntas que van apareciendo en pantalla. Una vez definido este protocolo de impresión, se puede guardar en disco (con un nombre), y utilizarse luego cuantas veces se quiera. Así ha quedado nuestro cuadro hasta este momento (ver figura 3).

Probemos ahora a variar, por ejemplo, las cifras de ventas de febrero y marzo, y comprobaremos que la variación se refleja inmediatamente en

El Calc Result, desvelado

las posiciones C13 y C14 correspondientes a Beneficios de esos meses, donde aparece ahora la nueva diferencia. Este es el aspecto más espectacular (y práctico) de **Calc Result** y en general de cualquier programa de cuadro de cálculo: una vez establecidas las relaciones entre las celdas, al variar los datos de partida podemos ver de manera inmediata su efecto sobre todo el cuadro (ver figura 4).

Si cuando copiamos las fórmulas con *Replicate*, hubiésemos elegido referencias absolutas en lugar de relativas, la fórmula reproducida en todas las columnas de Beneficios hubiera sido (B4 — B11), es decir, los datos de enero, y en este caso, los cambios hechos en la cifra de ventas de febrero y marzo no habrían tenido repercusión alguna en los correspondientes Beneficios.

¿Qué pasaría si...?

Tal como tenemos construido nuestro cuadro, caben muchas posibilidades de análisis y simulación. Por ejemplo, ¿cómo evolucionarían los beneficios si, manteniendo los gastos fijos, las ventas crecen un 1 por 100 mensual acumulativo?, ¿qué pasaría si los gastos de mano de obra se elevan un 7 por 100 en el segundo

FUNCIONES MATEMATICAS

GRUPO 1

COUNT: Da el número de posiciones que contienen una constante o fórmula válida (se excluyen NA o ERROR).

MAX: Selecciona el valor más elevado de un área dada.

MEAN: Calcula el promedio de un área dada.

MIN: Selecciona el valor más pequeño de un área dada.

STDDEV: Calcula la desviación standard de un área dada.

SUM: Suma los valores de un área dada.

***Estructura:* FUNCION (argumento, argumento, argumento...)**

***Ejemplo:* MEAN (A1: A5, B1: B5).**

GRUPO 2

NPV: Calcula el valor actual neto (Net Present Value).

***Estructura:* FUNCION (porcentaje, argumento).**

***Ejemplo:* NPV (.10,C1:G15).**

GRUPO 3

NA: Para una posición que carece de valor o una fórmula en que se incluye una referencia a una posición no válida.

PI: Da el valor π constante.

***Estructura:* FUNCION.**

***Ejemplo:* NA (Not Available = se desconoce).**

GRUPO 4

ABS, ARCCOS, ARCSIN, COS, EXP, FRAC, INST, LN, LOG10, RND, SQRT, SIN y TAN.

***Estructura:* FUNCION (expresión).**

***Ejemplo:* SIN (A1+1*10/B7).**

GRUPO 5

IF THEN ELSE.

***Estructura:* IF expresión THEN expresión ELSE expresión.**

***Ejemplo:* IF A1=10 THEN B1+2 ELSE 5.**

GRUPO 6

OR, AND y NOT.

***Estructura:* FUNCION dentro de IF THEN ELSE.**

***Ejemplo:* IF A1=10 OR C1=5 THEN B1+2 ELSE 5.**

Figura 7

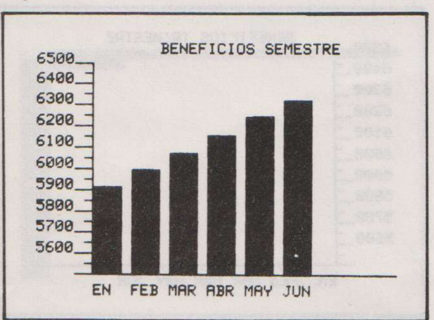
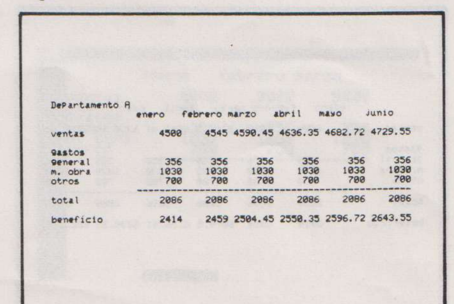


Figura 8



Departamento A

	enero	febrero	marzo	abril	mayo	junio
ventas	4500	4545	4590.45	4636.35	4682.72	4729.55
gastos						
general	356	356	356	356	356	356
m. obra	1030	1030	1030	1030	1030	1030
otros	700	700	700	700	700	700
total	2086	2086	2086	2086	2086	2086
beneficio	2414	2459	2504.45	2550.35	2596.72	2643.55

FORMAT COMMAND: C G M I $ L R *

C: Color	Selecciona color.
C: Color	Selecciona color.
G: Celda Global .	Activa formato global.
Gelda Global .	Activa formato normal.
M: Máximo	Activa modo de visualización en máxima precisión.
I: Entero	Activa modo de visualización entero.
$:	Visualización en dos decimales.
L: Izquierda	Activa tabulación a la izquierda.
R: Derecha	Activa tabulación a la derecha.
*:	Sustituye un número entero por el correspondiente número de asteriscos, tabulando por la izquierda.

trimestre?, ¿y si simultáneamente conseguimos reducir los gastos fijos?... Todas estas hipótesis tienen una fácil traducción en fórmulas, y su efecto, una vez introducidas, puede apreciarse de inmediato. Veamos por ejemplo la primera cuestión. En primer lugar "limpiamos" los valores de ventas de febrero a junio (con **f7 B**). Ahora, escribimos en ventas de febrero:

B4 = B4*1%

(O bien B4*1,01). A continuación, con el comando *Replicate*, reproducimos esta fórmula de C4 a G4 (marzo a junio), indicando como siempre referencias Relativas. He aquí el cuadro resultante (ver figura 5).

Calc Result posee una interesante función gráfica que permite representar en diagrama de barras cualquier fila o columna. Apliquemos esta función para visualizar de manera patente la evolución de nuestros Beneficios. Para ello situamos el cursor en esta fila (B13), y a continuación solicitamos de este modo la función gráfica:

F7 E G y R

El programa nos pide ahora la asignación de unas cotas superior e inferior para nuestro gráfico. Puesto que en nuestro caso el mínimo valor a representar es 5914 y el máximo 6322,08, le daremos como límites cifras próximas a ellos. Por ejemplo: **5500 Return 6500 Return**

En pantalla aparece ahora la petición de un título para el histograma. Para la prueba escribimos "BENEFICIOS DEL SEMESTRE", y a continuación pulsamos *Return*.

Para proceder a la impresión se pulsa la tecla **f6**, y antes de que comience, se tiene la opción de añadir dos hileras de texto (en el ejemplo hemos aprovechado una de ellas para poner los nombres de los meses). Y este es el resultado (ver figuras 6 y 7).

Páginas Múltiples

Calc Result está diseñado para manejar problemas de gestión de bastante entidad, que requieran gran cantidad de memoria. Para ello utiliza el disco, en cierto modo como memoria virtual o "Area de Trabajo", como confusamente la llama a veces el manual. Un cuadro de cálculo o fichero **Calc Result** puede constar de hasta 32 páginas, de las cuales sólo dos pueden residir simultáneamente en memoria.

Calc Result dispone de una serie de comandos para el manejo de estas páginas: copia, modificación, edición,... Veamos, por ejemplo, el caso de un análisis como el que hemos realizado, referido a dos secciones diferentes de la empresa, que llamaremos Departamento A y Departamento B. Vamos a utilizar para ambos los mismos datos numéricos con el fin de abreviar. Copiaremos primero la página 1 (en la que estábamos trabajando, y a la que añadiremos en A1 la etiqueta "Departamento A"), a la página 2. Basta para ello con la secuencia de órdenes:

f7 P C (llamada al comando de copia).

2 Return Y (especificación del destino de la copia).

La alternancia de página en pantalla (de las dos que coexisten en la memoria central), se consigue con la tecla **f1**. Presentemos la página 2 (el número de página aparece en el ángulo superior izquierdo, como P1, P2... P32). Situando el cursor en A1 escribimos "Departamento B". Estas serían nuestras dos páginas (ver figuras 8, 9, 10 y 11).

Ahora presentaremos ambas páginas en pantalla simultáneamente, lo

GLOBAL COMMAND: C F R

C: Anchura columna	Activa el ancho global en todas las columnas excepto en la columna de título protegida.
F: Formato	Activa el formato dado en todas las celdas.
R: Recalculo	Recalcula algunas páginas moviendo la columna más alta de una página a la columna Alfa de la página siguiente.

PAGE COMMAND: A C D E G N P R +

A: Adiciona	Páginas, chequeando las etiquetas y fórmulas iguales.
C: Copia	Copia una página a otra.
D: Borra	Borra página del área de trabajo.
E: Elimina	Area de trabajo.
G: Carga	Página desde el área de trabajo.
N: Negar	Cambia signos (+ y —) dentro una página.
P: Poner	Página dos a área de trabajo.
R: Renumerar ...	Página.
+:	Suma páginas, valores y fórmulas solamente.

Figura 9

```
P 1
  Departam
             enero    febrero marzo
  ventas      4500     4545 4590.45
  gastos
  general      356      356     356
  m. obra     1030     1030    1030
  otros        700      700     700
             ------------------------
  total       2086     2086    2086
  benefici    2414     2459 2504.45
```

que nos permitiría, por ejemplo, hacer un estudio comparativo de presupuestos para los dos departamentos. Para hacer esto, presentamos primero la página 1, y reducimos la anchura de sus columnas, para que podamos ver mayor número de ellas. Para esto hacemos:

f7 G C 6 y Return (La anchura de columnas es uno de los subcomandos de *Global*, y 6 es el ancho elegido).

Ahora situamos el cursor en D1 y llamamos a la orden *"Split"* (subcomando de Edición): **f7 E S**. Hecho esto, nos preguntará si queremos partir vertical u horizontalmente. Contestamos **V**, de vertical. La pantalla quedará partida, funcionando cada mitad de manera autónoma. El paso del cursor de una a otra mitad se consigue con la tecla **f2**. Pongámoslo en la parte derecha y llamemos a la página 2 sin más que pulsar la tecla de alternancia, esto es **f1**. Este será el aspecto de nuestra pantalla (previa reducción de la anchura de columnas

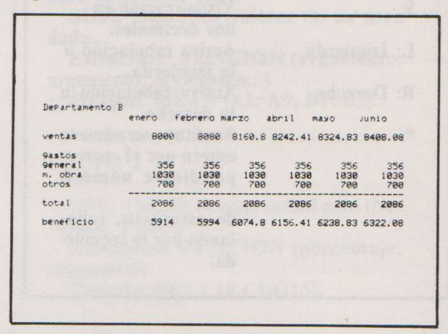
El Calc Result, desvelado

de P2, como hicimos antes con la otra) (ver foto 3).

La potencia de combinación de páginas de **Calc Result** es realmente grande, ya que la pantalla puede partirse en hasta cuatro ventanas independientes, cada una de las cuales puede mostrar parte de cuatro páginas diferentes. (Las páginas no residentes en memoria se llaman con el subcomando de página *Get*, que hace, a su vez, que la página reemplazada pase al disco). Esta capacidad ha hecho que algún observador haya calificado el programa como un **"VisiCalc tridimensional"**.

Otro de los subcomandos con interesantes posibilidades es la adición de páginas **(A)**. Cuando se utiliza, lo primero que hace es transferir a memoria la página de número más alto, que siempre es P32. Luego, cada una de las páginas incluidas en la suma (que hay que especificar previamente), se transfieren a ésta. Los valores de las celdas homólogas se suman, y en cuanto a los textos, si encuentra contenidos diferentes en posiciones idénticas, genera una advertencia al usuario por pantalla; si no se hacen cambios, prevalece el

Figura 10



Departamento B

	enero	febrero	marzo	abril	mayo	junio
ventas	8000	8080	8160.8	8242.41	8324.83	8408.08
gastos						
general	356	356	356	356	356	356
m. obra	1030	1030	1030	1030	1030	1030
otros	700	700	700	700	700	700
total	2086	2086	2086	2086	2086	2086
beneficio	5914	5994	6074.8	6156.41	6238.83	6322.08

texto de la página que entró primero.

En nuestro caso, la llamaríamos así:

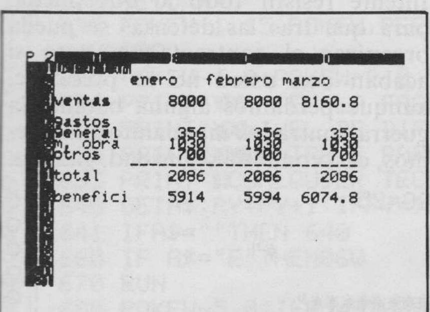
f7 P A (llamada al subcomando de Adición).

N (especifica que no queremos sumar todas las páginas).

1 Return 2 Return (relación de páginas a sumar).

Y éste sería el resultado en P32 (después de poner la etiqueta "GENERAL..." en A1) (ver figura 12):

Figura 11



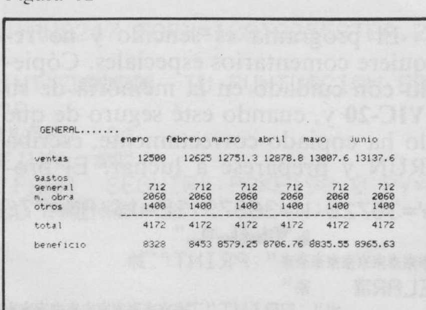
P 2

	enero	febrero	marzo
ventas	8000	8080	8160.8
gastos			
general	356	356	356
m. obra	1030	1030	1030
otros	700	700	700
total	2086	2086	2086
benefici	5914	5994	6074.8

Son muchas las posibilidades y opciones que no hemos ejercitado en la creación de nuestro pequeño modelo. Esta presentación de **Calc Result** no pretende más que dar a conocer las grandes posibilidades que tienen este tipo de programas.

Los programas de cuadros de cálculo ganan en fertilidad e interés cuanto más familiarizado se está con ellos. Son innumerables los casos a que pueden aplicarse, pero podría

Figura 12



GENERAL.......

	enero	febrero	marzo	abril	mayo	junio
ventas	12500	12625	12751.3	12878.8	13007.6	13137.6
gastos						
general	712	712	712	712	712	712
m. obra	2060	2060	2060	2060	2060	2060
otros	1400	1400	1400	1400	1400	1400
total	4172	4172	4172	4172	4172	4172
beneficio	8328	8453	8579.25	8706.76	8835.55	8965.63

decirse, en general, que juegan con ventaja sobre los programas dedicados, en aquellas aplicaciones que varían frecuentemente de una a otra utilización dada la facilidad con que pueden modificarse, y la observación inmediata que ofrecen del efecto de los cambios. La declaración del impuesto sobre la renta, por hablar de algo que afecta a prácticamente todo el mundo, es algo que, de estar resuelto con un programa, habría que rehacer prácticamente todos los años, mientras que con **Calc Result**, la clase de cambios que suelen aparecer (porcentajes modificados, algún concepto nuevo, etcétera), pueden incorporarse con facilidad.

Y no sólo aplicaciones de este tipo. A la vista del cuadro de funciones matemáticas disponibles, es claro que **Calc Result** resultará también una herramienta interesante en problemas de cálculo y estadística.

Gumersindo García

VIC - 20

Comando estelar

Los marcianitos, ya se sabe, siempre nos están atacando. Desde que se inventaron los juegos de ordenador, no han dejado de atacarnos de todas las formas imaginables. En esta continuada guerra intergaláctica, muchas naves terrestres han sido destruidas, pero los marcianitos, no han salido mucho mejor parados, también han sufrido muchas bajas. En este programa, la guerra continúa. Esta vez, las naves invasoras, intentan destruir las defensas terrestres. Usted está al mando de una flotilla de defensa compuesta por tres naves, equipadas con rayos laser, y su misión es resistir el mayor tiempo posible el ataque exterior, destruyendo cualquier nave invasora que se le acerque.

El programa es sencillo y no requiere comentarios especiales. Cópielo con cuidado en la memoria de su **VIC-20** y, cuando esté seguro de que lo ha copiado correctamente, escriba RUN y prepárese a luchar. El programa cuenta con 10 niveles de dificultad que determinan la velocidad de ataque de las naves enemigas. Si los invasores consiguen destruir sus tres naves, el juego habrá terminado y el ordenador le dará su puntuación. Intente resistir todo lo que pueda, para que tras las defensas se pueda organizar el contraataque, pero si acaban con usted no se preocupe, aunque perdamos alguna batalla, la guerra contra los marcianitos la tenemos, casi con toda seguridad, ganada.

```
100 POKE650,128:POKE198,0:Y=30720:W=36879:SE=16:AM=7768:CC=25
111 POKEW,12:PRINT"QQQQQQ                    "
112 PRINT"QQQ************************":PRINT"*                          *"
113 PRINT"* COMANDO ESTELAR    *"
114 PRINT"*                    *":PRINT"************************"
115 FORC=1TO4000:NEXT:PRINT"INSTRUCCIONES:-"
116 PRINT" F1 MUEVE ARRIBA, F7  MUEVE ABAJO,F DISPARA"
117 PRINT" DISPARA A LAS NAVES ENEMIGAS,SINO DESTRUIRAN TUS DEFENSAS"
118 PRINT"DISPONES DE 3 NAVES"
119 PRINT" BUENA SUERTE"
120 PRINT"ELIGE TU NIVEL DE DIFICULTAD(0-9),EL 0 ES EL MAXIMO":PY=0
122 GETO$:PY=PY+1:IFPY=>2000 THENO$="9"
123 IFO$>"9"ORO$<"0"THEN122
129 D2=VAL(O$)*10:POKEW,8:PRINT"":A=7702:B=7724:E=B:F=7703:G=7352:H=7766:I=B+2
:U=7919
130 FORC=1TO9:A=A+44:FORD=1TO8:A=A+2:POKEA,91:POKEA+30720,5:NEXTD:A=A-16:NEXTC
140 FORC=1TO17:E=E+44:E=E-16:NEXTC
150 FORC=1TO9:F=F+44:FORD=1TO7:F=F+2:POKEF,64:NEXTD:F=F-14:NEXTC
160 Z=121:FORC=0TO1:G=G+374
170 FORD=1TO15:POKEG,Z:G=G+1:NEXTD:G=G-15:G=G+22:Z=Z-1:NEXTC:G=G+7:Z=Z-1
190 FORC=1TO17:POKEH,124:POKEH+30720,6:H=H+22:NEXTC
195 FORAZ=38463TO38843STEP22:POKEAZ,3:NEXTAZ
200 Z=55:FORC=1TO8:POKEI,Z:POKEI+30720,Z-47:I=I+2:Z=Z-1:NEXT:POKE38446,5:S=3:S1=
0:P=7
210 POKE7919,60:GOSUB842
230 GETK$:IFK$=""THEN290
250 IFK$=""THEN310
260 IFK$="F"THEN330
261 POKE37145,127:XS=PEEK(37137):IF((XSAND4)=0)<>0THEN290
264 IF-((XSAND32)=0)=1THEN330
265 IF-((XSAND8)=0)<>0THEN310
266 POKE37154,255:GOSUB540:GOTO230
290 IFT=>3THENGOSUB540:GOTO230
300 POKEU,32:T=T+1:U=U-44:POKEU,60:GOTO360
310 IFT=<-4THENGOSUB540:GOTO230
320 POKEU,32:T=T-1:U=U+44:POKEU,60:GOTO360
330 M=U-1:FORQ=15TO0STEP-1:POKEW-5,128:POKEW-1,Q:POKEM,32:IFM=J+4THENS1=S1+1:GOT
O680
340 M=M-1:POKEM,45:NEXTQ:POKEW-5,0
```

```
360 POKEM,32:GOSUB540:GOSUB540:GOTO230
500 N=30720:D=0:E=0:D=INT(RND(1)*16):WW=D/2:IFWW=INT(WW)THEN530
520 GOTO500
530 F=D*22:J=(F+AM)-1
531 GOSUB900:RJ=INT(10*RND(0)+1):IFRJ=8THEN GOSUB910
540 J=J+1:POKEJ-1,32:POKEJ,R(1):POKEJ+1,R(2):POKEJ+2,R(3):POKEJ+3,R(4):POKEJ+N+3
,1
549 POKEJ+Y,3:POKEJ+Y+1,2:POKEJ+Y+2,5:POKEJ+Y+3,7:FORX=1TOD2-S1:NEXT
560 E=E+1:IFE=SETHENS=S-1:POKEW,PEEK(W)AND247:POKEJ,32
561 IFE=SETHENPOKEJ+1,32:POKEJ+2,32:POKEJ+3,32:GOTO760
570 IFS=0THEN590
580 RETURN
590 POKEW-1,15:POKEW,PEEK(W)AND247:FORX=128TO255STEP.2:POKEW-3,X:NEXTX:POKEW,PEE
K(W)OR8
610 POKEW-1,0:POKEW-2,0:PRINT"█████  TU PUNTUACION ES  "S1
620 PRINT"██   EN LA FASE "P
625 PRINT"██NIVEL DE DIFICULTAD "O$
630 PRINT"███ENTRE E PARA FINAL ██"
635 PRINT"█CUALQUIER TECLA PARA  SEGUIR":POKE198,0:PY=0
640 GETA$:PY=PY+1:IFPY=>2000 THENA$="U"
641 IFA$=""THEN 640
660 IF A$="E"THEN860
670 RUN
680 POKEW-5,0:IFRJ=8THENS1=S1+4
681 M=0:L=0:V=2:POKEJ+N,V:POKEJ+1+N,V:POKEJ+2+N,V:POKEJ+3+N,V
690 POKEW-2,220:FORL=15TO0STEP-.15:POKEW-1,L:NEXTL:SH=SH+1:POKEW-2,0:POKEW-1,0
730 FORC1=0TO3:POKEJ+C1,32:NEXTC1
740 IFCC=SHTHENS=S+1:CC=CC+25:P=P-1:AM=AM+2:SE=SE-2:GOSUB800
760 IFE=SETHENGOSUB820
770 GOSUB842:POKEW,PEEK(W)OR8:GOTO230
800 FORX=200TO255:POKEW-3,X:NEXT
820 POKEW-1,15:POKEW-2,220:FORL=15TO0STEP-.1:POKEW-1,L:NEXT:POKEW-2,0:POKEU,60:R
ETURN
842 PRINT"██NAVES█"S:PRINT"████████████████████FASE█"P"█PUNTOS█"S1:GOSUB500:RET
URN
860 PRINT"████ ADIOS!":END
900 R(1)=95:R(2)=98:R(3)=121:R(4)=70:RETURN
910 R(1)=32:R(2)=62:R(3)=232:R(4)=223:RETURN
```

COMMODORE 64

Cavernas de hielo



Los programas de juegos de ordenador, están alcanzando unas enormes cotas en cuanto a imaginación, complejidad y sofisticación. Este programa, por el contrario, se puede considerar un clásico, pero resulta igualmente apasionante.

Se trata de un laberinto tridimensional excavado en el hielo. Es tridimensional en cuanto a su presentación en pantalla, pero también lo es en cuanto a la estructura misma del laberinto. Esta estructura, excavada enteramente en el hielo, consiste en cinco plantas cuadradas, una encima de otra, cada una de ellas dividida en veinticinco cámaras diferentes, organizadas en una matriz de 5 por 5. Cada una de las cámaras tiene un total de seis posibles salidas: norte, sur, este, oeste, arriba y abajo. Sólo una de las salidas, en una de las cámaras, de uno de los pisos, conduce al exterior.

El programa, que lleva incluidas las instrucciones para jugar, le permite moverse por todo el laberinto. También le permite, estando en una cámara, determinada, cambiar su punto de vista y situarse de frente a una cualquiera de las cuatro paredes. También hay un contador de tiempo que le dirá cuánto tiempo lleva dando vueltas por la estructura de hielo; procure salir deprisa o se quedará congelado. Una vez que alcance la salida, podrá escoger entre volver al mismo laberinto desde el mismo punto de partida y tratar de salir en menos tiempo, o bien, empezar de nuevo en un nuevo laberinto aleatorio.

```
1 X=RND(-π)
10 R$="▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐▐|"
20 D$="▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓"
25 PRINT"▔"
90 GOSUB 2000
100 DIMFC(5,7),FC$(5)
105 FC$(1)="NORTE":FC$(2)=" SUR ":FC$(3)=" ESTE":FC$(4)="OESTE"
110 FORB=1TO4:FORI=1TO6:READFC(B,I):NEXT:NEXT
115 GOTO155
120 PRINT"▔ ________________________▓▌/▓▌/▓▌/▓▌/"
121 PRINT"▓▓▌▓╲▓╲▓╲________________"
122 PRINT"▓▓";:FORI=1TO18:PRINT" ▓▌";:NEXT:PRINT"▔▌/▔/▔/▔/▔";
123 PRINT"| ";:FORI=1TO9:PRINT"▔▌ ";:NEXT
124 PRINT"▓";LEFT$(D$,19);" ▓                    ▀ ▌▔▌╲▔▌╲▔▌╲▔▌╲"
```

```
125 PRINT"▒"LEFT$(D$,15);LEFT$(R$,5);"‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾‾"
126 PRINT"▒";LEFT$(R$,21);:FORI=1TO18:PRINT"▒ ▌▌";:NEXT
127 PRINT"▒";LEFT$(R$,17);LEFT$(D$,5);:FORI=1TO10:PRINT"| ▒▌";:NEXT
129 RETURN
130 PRINT"▒";LEFT$(R$,8);"▒______▒██████N    /▒█████┌‾‾‾":RETURN
135 PRINT"▒";LEFT$(D$,16);LEFT$(R$,9);"_____▒██████/    N▒██████┌‾‾‾‾‾":RETURN
140 PRINT"▒";LEFT$(D$,7);"▌▌";:FORI=1TO11:PRINT" ▒▌";:NEXT:PRINT"▌┌┬┐└┘"
141 PRINT"▒";LEFT$(D$,9);"▌▌┌┐";:FORI=1TO5:PRINT"▒▌|";:NEXT
142 PRINT"▒";LEFT$(D$,7);"▌▌N▒\":RETURN
145 PRINT"▒";LEFT$(D$,7);LEFT$(R$,9);"_____▒█████ N__/ ";:FORI=1TO5
146 PRINT"▒███████| | | | ";:NEXT:PRINT"▒███████ /‾‾N ":RETURN
150 PRINT"▒";LEFT$(D$,7);LEFT$(R$,19);"/ ▒███/ |▒████┐ ";:FORI=1TO5
151 PRINT"▒███    |";:NEXT:PRINT"▒███└_┘ ▒▌ ▒▌ ":RETURN
155 DIMS$(6,6)
165 FORA=1TO5:FORX=1TO5:FORY=1TO5
170 IFA<>5ANDRND(1)<.8THENS$(X,A)=S$(X,A)+"O":GOTO180
175 S$(X,A)=S$(X,A)+"X"
180 IFMID$(S$(X,A-1),(Y-1)*6+1,1)="O"THENS$(X,A)=S$(X,A)+"O":GOTO190
185 S$(X,A)=S$(X,A)+"X"
190 IF Y-2<0GOTO200
195 IFMID$(S$(X,A),(Y-2)*6+4,1)="O"THENS$(X,A)=S$(X,A)+"O":GOTO205
200 S$(X,A)=S$(X,A)+"X"
205 IFY<>5ANDRND(1)<.8THENS$(X,A)=S$(X,A)+"O":GOTO215
210 S$(X,A)=S$(X,A)+"X"
215 IFX<>5ANDRND(1)<.8THENS$(X,A)=S$(X,A)+"O":GOTO225
220 S$(X,A)=S$(X,A)+"X"
225 IFMID$(S$(X-1,A),(Y-1)*6+5,1)="O"THENS$(X,A)=S$(X,A)+"O":GOTO235
230 S$(X,A)=S$(X,A)+"X"
235 NEXT:PRINT"*";:NEXT:NEXT
240 X=INT(RND(1)*3)+2:Y=INT(RND(1)*3)+2:A=INT(RND(1)*3)+2
245 RD=INT(RND(1)*6)+1:ONRDGOTO250,255,260,265,270,275
250 A=5:P1$=LEFT$(S$(X,A),(Y-1)*6):L=29-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
251 S$(X,A)=P1$+"O"+P2$:GOTO290
255 A=1:P1$=LEFT$(S$(X,A),(Y-1)*6+1):L=29-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
256 S$(X,A)=P1$+"O"+P2$:GOTO290
260 Y=5:P1$=LEFT$(S$(X,A),(Y-1)*6+3):L=29-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
261 S$(X,A)=P1$+"O"+P2$:GOTO290
265 Y=1:P1$=LEFT$(S$(X,A),(Y-1)*6+2):L=29-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
266 S$(X,A)=P1$+"O"+P2$:GOTO290
270 X=5:P1$=LEFT$(S$(X,A),(Y-1)*6+4):L=29-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
271 S$(X,A)=P1$+"O"+P2$:GOTO290
275 X=1:P1$=LEFT$(S$(X,A),(Y-1)*6+5):L=29-LEN(P1$):P2$=RIGHT$(S$(X,A),L)
276 S$(X,A)=P1$+"O"+P2$:GOTO290
290 PRINT:PRINT: PRINT"PULSA ▒RETURN▒ PARA EMPEZAR"
300 GETC$:IFC$=""GOTO300
1000 X=INT(RND(1)*5)+1:Y=INT(RND(1)*5)+1:A=INT(RND(1)*5)+1
1005 SX=X:SY=Y:SA=A
1010 FC=1:TI$="000000":GOTO1220
1020 PRINT"▒";LEFT$(D$,20);LEFT$(R$,16)"▒";FC$(FC):A$="":D=0
1030 TX$=TI$:TP$=MID$(TX$,2,1)+":"+MID$(TX$,3,2)+":"+RIGHT$(TX$,2)
1040 PRINT"▒";LEFT$(D$,20);" ▒TIEMPO=";TP$;" "
1050 GET A$
1060 IFA$="A"THEND=1
1070 IFA$="B"THEND=2
1080 IFA$="N"THEND=3
1090 IFA$="S"THEND=4
1100 IFA$="E"THEND=5
```

Juegos

Viene de la página anterior

```
1110 IFA$="O"THEND=6
1120 IFA$="V"GOTO1280
1130 IFD=0GOTO1020
1140 IFMID$(S$(X,A),(Y-1)*6+D,1)<>"O"THENGOTO1020
1150 ONDGOTO1160,1170,1180,1190,1200,1210
1160 A=A+1:GOTO1220
1170 A=A-1:GOTO1220
1180 Y=Y-1:GOTO1220
1190 Y=Y+1:GOTO1220
1200 X=X+1:GOTO1220
1210 X=X-1
1220 IFX>5ORX<1ORY>5ORY<1ORA>5ORA<1THEN PRINT"GANASTE. OTRO JUEGO?":GOTO3000
1230 GOSUB120
1240 FORII=1TO6:IFMID$(S$(X,A),(Y-1)*6+II,1)="X"THENNEXT:GOTO1020
1250 R=FC(FC,II)+1
1260 ONRGOSUB125,130,135,140,145,150
1270 NEXT:GOTO1020
1280 PRINT"■";LEFT$(D$,22);"NUEVA VISTA, N,S,E,O";
1281 GETC$:IFC$=""GOTO1280
1282 IFC$<>"N"ANDC$<>"S"ANDC$<>"E"ANDC$<>"O"GOTO1281
1283 PRINTC$:IFC$="N"THENFC=1
1284 IFC$="S"THENFC=2
1285 IFC$="E"THENFC=3
1286 IFC$="O"THENFC=4
1287 GOTO1220
2000 PRINTLEFT$(D$,8);LEFT$(R$,5);"■CUEVAS DE HIELO■"
2001 PRINT"■■■QUIERES":INPUT"INSTRUCCIONES   ■■■■";Y$
2002 IFLEFT$(Y$,1)<>"S"THENGOTO2100
2010 PRINT"■EL OBJETO DEL ■JUEGO■":PRINT"ES ENCONTRAR LA SALIDA"
2011 PRINT"DE UN LABERINTO ":PRINT"CUBICO DE 5X5X5.
2012 PRINT"EN UNA DE LAS HABITACIONES":PRINT"HAY UNA SALIDA.
2013 PRINT:PRINT"DEBES TRATAR DE ENCONTRARLA":PRINT"EN EL MINIMO TIEMPO POSIBLE.
2014 PRINT"LOS COMANDOS SON :"
2020 PRINT"■A■ - ARRIBA; ■S■ - SUR;":PRINT"■B■ - ABAJO ; ■E■ - ESTE;
2030 PRINT"■N■ - NORTE ; ■O■ - OESTE"
2040 PRINT"■V■ PARA CAMBIAR EL PUNTO DE VISTA.
2050 PRINT:PRINT"PULSA ■RETURN■ PARA EMPEZAR.
2051 GETC$:IFC$=""GOTO2051
2060 PRINT"■■V■ APARECERA SEGUIDO DE":PRINT"UNO DE LOS COMANDOS
2062 PRINT"SEGUN LA VISTA QUE DESEES":PRINT"PULSA SOLO UNA":PRINT"TECLA Y ";
2063 PRINT"■RETURN■"
2100 PRINT:PRINT"ESPERA UNOS 30":PRINT"SEGUNDOS MIENTRAS HAGO"
2101 PRINT"EL LABERINTO"
2102 RETURN
3000 PRINT"    ■S■ OR ■N■?"
3010 GETC$:IFC$=""GOTO3010
3020 IFC$<>"S"ANDC$<>"N"GOTO3010
3030 IFC$="N"THENSTOP
3032 PRINT"MISMO LABERINTO ■M■ O
3033 PRINT"UNO NUEVO ■N■?"
3034 GETC$:IFC$=""GOTO3034
3035 IF C$<>"M"ANDC$<>"N"GOTO3034
3036 IFC$="N"GOTO165
3040 X=SX:Y=SY:A=SA:GOTO1010
4000 DATA1,2,4,0,5,3,1,2,0,4,3,5,1,2,3,5,4,0,1,2,5,3,0,4
```

Ocurre a menudo que, después de teclear cuidadosamente un largo programa, del que tenemos el listado; después de dar un suspiro al haber acabado la ingrata labor y después de haber escrito la palabra mágica RUN, pulsamos la tecla RETURN y ocurre lo que era de temer: O bien aparecen frases indicándonos errores de sintaxis, o sentencias no definidas o algo por el estilo, o bien no aparece ninguna de estas frases, pero el programa simplemente no funciona.

En tal caso se hace necesario listar el programa y repasar cuidadosamente todas las instrucciones, hasta encontrar aquella en la que hemos cambiado una letra, o nos hemos comido un signo, quizás nos faltan o sobran ceros en alguna constante.

Para este ejemplo, como también para muchos otros, sobre todo en lo relacionado con la detección de errores puede sernos de utilidad esta corta rutina, que nos permite listar cualquier programa, línea por línea y en ambas direcciones, es decir, pasando a la línea siguiente o volviendo a la línea anterior a partir de una dada, utilizando únicamente las teclas + y —.

Para ello basta añadir, al final del programa esta rutina, y escribir RUN 9999 con lo que en la pantalla aparecerá listada la primera línea del programa.

Para listar las siguientes líneas pulsar + y para volver atrás pulsar —. Para los que quieran saber cómo funciona ahí va la explicación:

10000	inicializa ADDRESS al comienzo del BASIC.
10010	actualiza LINE al número de línea siguiente o anterior, dependiendo de la entrada en 11000.
10020	limpia la pantalla y escribe LIST y LINE.
10030	actúa mediante POKEs en el *buffer* de teclado, introduciendo las órdenes HOME, CURSOR abajo, color, RETURN, CURSOR abajo, RETURN.
11000-11020	bucle para entrada de + ó —.
12000	busca el principio de línea.
12010-13000	actualiza ADDRESS.
13005	verifica si empieza una nueva línea.
13010	actualiza ADDRESS para la última línea.

VIC - 20

```
9999 REM**LISTADOS +/-*
10000 ADDRESS=PEEK(44)*256+PEEK(43)-1
10010 LINE=PEEK(ADDRESS+3)+PEEK(ADDRESS+4)*256
10020 PRINT"GOTO11000":PRINT"LIST";LINE;
10030 POKE631,19:POKE632,17:POKE633,31:POKE634,13:POKE635,19:POKE636,13:POKE198,
6:END
11000 IFPEEK(197)=5THEN12000
11010 IFPEEK(197)=61THEN13000
11020 GOTO11000
12000 IFPEEK(ADDRESS+5)<>0THENADDRESS=ADDRESS+1:GOTO12000
12010 ADDRESS=ADDRESS+5:GOTO10010
13000 ADDRESS=ADDRESS-1
13005 IFPEEK(ADDRESS)=0ANDPEEK(ADDRESS-4)<>0ANDPEEK(ADDRESS-3)<>0THEN10010
13010 GOTO13000
```

El buffer de teclado resulta muy útil cuando tecleemos a gran velocidad. Pero también puede resultarnos contraproducente en un programa si antes de una instrucción GET no lo vaciamos. Podría ser que pulsemos accidentalmente una tecla "no deseada" durante la ejecución de nuestro programa.

Antes de la instrucción GET deseada, y según las pretensiones de nuestro programa, conviene vaciar el buffer del teclado.

Una forma sería:

```
10 GET A$: IF A$>"" THEN 10
```

O bien:

```
10 FOR I=1 TO 10:GET A$:NEXT I
```

Intente ejecutar el programa de ejemplo introduciendo la instrucción de vaciado y sin ella, pulsando algunas teclas antes de que aparezca el mensaje en la pantalla. ("PULSA UNA TECLA").

```
0 FOR J=1 TO 3000:NEXTJ
5 REM***0 SIMULA UN PROGRAMA EN CURSO***
10 REM***AQUI VA LA INSTRUCCION DE VACIADO***
20 PRINT"PULSA UNA TECLA"
30 GET A$:IFA$=""THEN 30
40 PRINTA$
50 END
```

Juegos

Tecla Fast

Si sus dedos no son los bastante ágiles sobre el teclado, y cada vez que quiere pasar un programa de más de 50 líneas, se ve obligado a cancelar todas sus citas de la tarde, lo que usted necesita es este sencillo y entretenido programa.

El programa consiste sencillamente en que, sobre la pantalla en blanco de su televisor, aparecerán letras al azar. Usted dispone de unos segundos para pulsar la tecla correspondiente. Si falla, porque pulsa una tecla incorrecta o porque excede del tiempo permitido, sonará un tono y se le avisará del fallo, no incrementándose su puntuación.

El tiempo de que dispone para pulsar la tecla correcta, viene determinado por la sentencia 140 FOR I=1 TO 500. Si en lugar de 500 elige otro valor más pequeño, como por ejemplo, 200 ó 100, tendrá que ser mucho más rápido o no acertará ni una sola vez.

COMMODORE 64

PROGRAMA COMPROBADO
commodore Magazine

```
1 PRINT"▒■"
5 GOSUB310
7 POKE53280,6:POKE53281,3
8 PRINT"▒"
10 PRINTSPC(7) "****** TECLAFAST ******"
11 PRINTSPC(7) "▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬"
12 PRINT:PRINT:PRINT
15 PRINT"  PULSA LA TECLA ......MAX PTOS 25"
16 FORT=1TO5000
17 NEXTT
20 LET H=0
30 FOR T=1 TO 25
40 FOR I=1TOINT(RND(1)*300+1000)
50 NEXT I
60 LET A=INT(RND(1)*40)
70 LET D=INT(RND(1)*25)
80 LETP$=CHR$(INT(RND(1)*26)+65)
90 PRINT"▒"
100 FORJ=0TOD
110 PRINT
120 NEXTJ
130 PRINTTAB(A);"▒"P$
140 FORI=1TO500
150 GETR$
160 IFR$=P$THEN GOTO210
170 IFR$<>""THEN GOTO190
180 NEXT I
190 PRINT"FALLO":GOSUB500
200 GOTO230
210 PRINT"     VALE":GOSUB400
220 LETH=H+1
230 NEXT T
240 PRINT"▒"
250 PRINT"▒▒▒▒▒▒▒▒▒▒▒▒TUS PUNTOS SON ";H;"/25"
260 PRINT"▒▒▒OTRA VEZ? S/N"
270 INPUTB$
280 IF B$="S"GOTO8
290 IF B$="N"GOTO300
300 END
310 PRINT"▒▒▒▒EL PROGRAMA MUESTRA UNA LETRA
    AL AZAR ENLA PANTALLA."
320 PRINT"▒▒PULSA LA LETRA ANTES DE QUE
    DESAPAREZCA,PERO NO DESPUES DEL PITIDO."
330 PRINT"▒▒TIENES SOLO UN INTENTO POR CADA LETRA."
340 PRINT"▒▒▒▒**PULSA CUALQUIER TECLA PARA
    CONTINUAR**"
```

```
350 GETK$:IFK$=""GOTO350
360 RETURN
400 S=54272:POKES+24,15:POKES+5,100:POKES+1
    ,25:POKES+0,177:POKES+4,17
410 FORP=1TO100:NEXT
420 POKES+4,0:POKES+5,0:POKES+6,0
430 RETURN
500 S=54272:POKES+24,15:POKES+5,100:POKES+1
    ,12:POKES+0,216:POKES+4,17
510 FORP=1TO100:NEXT
520 POKES+4,0:POKES+5,0:POKES+6,0
530 RETURN
```

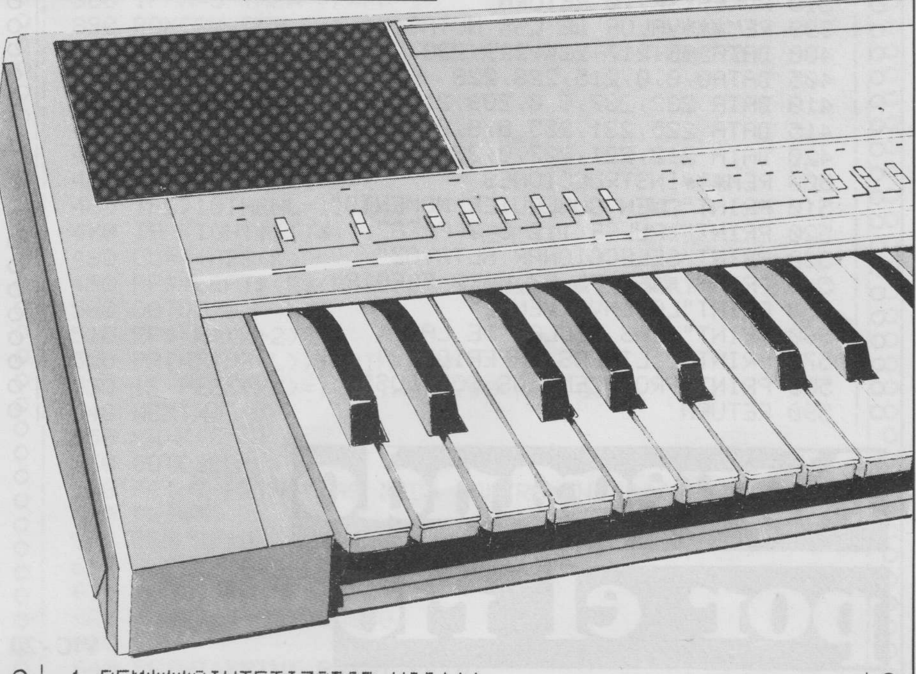
Sintetizador Vic

Una de las características interesantes de los actuales microordenadores es su capacidad para producir diversos sonidos. El Commodore 64, por ejemplo, lleva incorporado un circuito integrado específico, que hace de este microordenador un auténtico sintetizador musical con amplias posibilidades. El VIC-20, por su parte, posee unas capacidades de todo tipo de ruidos y sonidos, a costa, eso sí, de un mayor esfuerzo por parte del programador.

En el manual de usuario del VIC-20, existe un programa para transformar el teclado, en un teclado para la producción de sonidos. Este programa, hace lo mismo, pero permite seleccionar la envolvente correspondiente a cada uno de los tres altavoces y modificarla a voluntad durante la ejecución del programa. Las dos hileras inferiores del teclado, se transforman de este modo, en las teclas de un sintetizador.

Aunque corto, este programa permite experimentar con diversas envolventes, y proporciona una idea de las capacidades musicales del VIC-20.

VIC - 20



```
1 REM***SINTETIZADOR VIC***
5 PRINT"♡"
7 GOSUB500
10 VO=36878:DIM P(28):DIM S(3):N=0:EN=1:SP=2
15 FORI=0TO28:READP(I):NEXT
20 S(1)=36874:S(2)=36875:S(3)=36876
25 N=PEEK(197):IFN=64 THEN25
30 IFN=>17ANDN=<45THENQ=P(N-17):GOTO70
35 IFN=0 THEN SP=1
40 IFN=56 THEN SP=2
45 IFN=1 THEN SP=3
50 IF N=47 THEN EN=1
55 IF N=55 THEN EN=2
60 IF N=63 THEN EN=3
```

Juegos

Viene de la página anterior

```
65 GOTO25
70 ON EN GOSUB 200,250,300
75 GOTO25
199 REM***ENVOLVENTE 1***
200 POKEVO,15:POKES(SP),Q:POKES(3),Q
205 IF PEEK(197)=N THEN205
210 POKES(SP),0:POKE S(3),0:RETURN
249 REM***ENVOLVENTE 2***
250 POKES(SP),Q:FORV=2 TO 15 STEP.5:POKEVO,V:NEXT
255 IF PEEK(197)=N THEN255
260 FORV=15 TO 0 STEP-1:POKEVO,V:NEXT:
    POKES(SP),0:RETURN
299 REM***ENVOLVENTE 3***
300 POKEVO,5:POKES(SP),Q:POKEVO,15
305 IF PEEK(197)=N THEN 300
310 POKES(SP),0:RETURN
399 REM***VALOR DE LAS NOTAS***
400 DATA208,217,224,229,233,238
405 DATA0,0,0,215,223,228
410 DATA 232,237,0,0,209,219
415 DATA 225,231,235,0,0,0
420 DATA 212,221,227,0,236
500 REM***INSTRUCCIONES
510 PRINT"❏❋EN CUALQUIER MOMENTO"
520 PRINT"❋PULSE 1,2 O 3 PARA"
530 PRINT"SELECCIONAR ALTAVOZ"
540 PRINT"❋❋F3,F5,F7 PARA ELEGIR"
550 PRINT"LA ENVOLVENTE"
560 PRINT"❋LAS TECLAS DE LAS"
570 PRINT"2 LINEAS INFERIORES"
580 PRINT"PRODUCEN LOS SONIDOS❋"
590 RETURN
```

Navegando por el rio

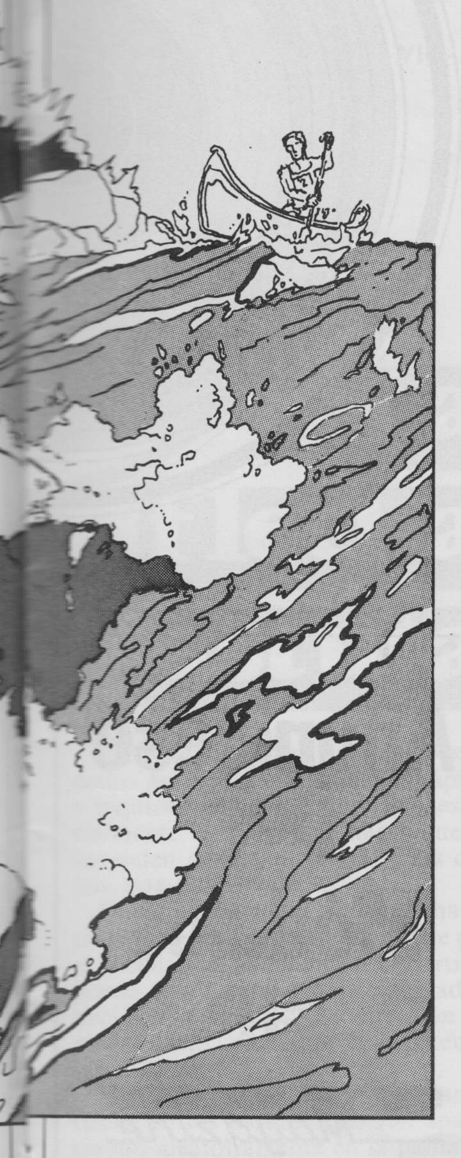
VIC - 20

```
100 REM
110 REM
120 REM
130 VR=PEEK(648)*256
140 KR=38400:IFVR<>7680THENKR=37888
150 PK=7680+296:POKE36879,25:POKE36869,240
160 PRINT"❏❋**NAVEGANDO POR EL"
163 PRINT
165 PRINT"MISISSIPY***"
170 PRINT"❋MONTA EN TU BARCO Y"
180 PRINT" NAVEGA POR EL RIO VIC"
190 PRINT"❋SE MANEJA CON + Y -"
200 PRINT"❋     +=IZQUIERDA"
```



Póngase la gorra de capitán y prepárese a manejar un bote por un río lleno de cambios de dirección y de puntiagudas rocas en medio de la corriente, que deberá sortear. Lleno, también, de manchas de aceite debidas a los vertidos industriales que, si no tienen cuidado con ellas, harán que pierda el control de su bote.

El programa cuenta con varios niveles de dificultad, que se traducen en la mayor o menor velocidad del bote por el río. Así que, practique y



podrá bajar el río más y más deprisa cada vez.

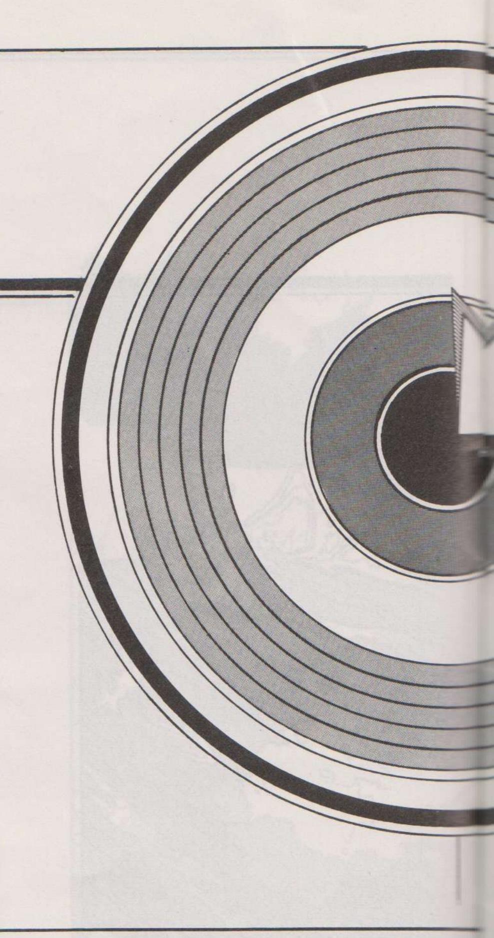
Gobernar el bote es muy sencillo, sólo tiene que usar las teclas + y — para desplazarse a izquierda y derecha respectivamente.

Al terminar, el programa le dirá los km. recorridos por el río y la velocidad de navegación.

¡Cuidado! Esta no es la piragua de Guillermo Cubillo. Además: Hay caimanes.

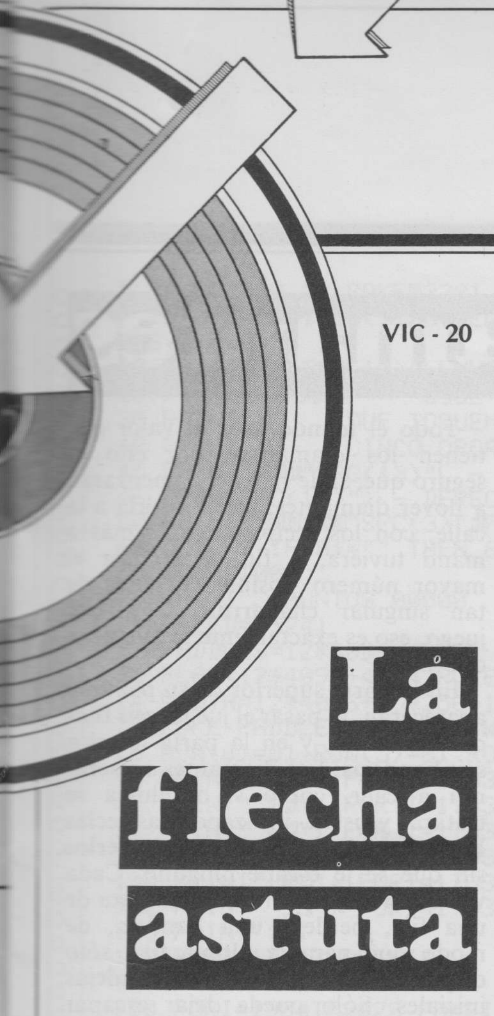
```
210 PRINT"▣         -=DERECHA"
220 R$(1)="▣◤              ▀◤"
230 R$(2)="▣H            H▀"
240 R$(3)="◥▣              ◥▀"
250 L=5
260 KW=0
270 GOSUB640
280 MILES=0
290 FORX=1TO15:PRINTTAB(5)+R$(2):NEXTX
300 K=INT(RND(9)*3)-1
310 IFKW=K THEN GOTO340
320 IFKW=1 THEN L=L+1:GOTO340
330 IFKW<1 ANDK=1 THENL=L-1
340 LAST=INT(RND(9)*9)
350 FORX=1TOLAST
360 AA=PEEK(197)
370 IFAA=61THENPK=PK+1
380 IFAA=5 THEN PK=PK-1
390 POKEPK,209
400 IFPEEK(PK-1)=32 OR PEEK(PK+1)=32 THEN570
410 MILES=MILES+100-XX%
420 FORY=1TOXX%:NEXTY
430 L=L+K
440 POKEPK,160
450 IFL<1THENL=1:K=0
460 IFL>10THENL=11:K=0
470 IF RIGHT$(TI$,1)<>"0"THEN510
480 LL=4+ABS(K)
490 RP$=LEFT$(R$(K+2),LL)+"▀▒▣"+RIGHT$(R$(K+2),5)
500 GOTO520
510 RP$=R$(K+2)
520 PRINTTAB(L);:PRINTRP$
530 IF PEEK(PK)=102 THEN 790:REM OIL
540 NEXTX
550 KW=K
560 GOTO300
570 PRINT:PRINT"HAS DADO CONTRA UNA ROCA !!!"
580 PRINT
590 GOSUB710
600 GETA$:IFA$=""THEN600
610 IFA$="N"THEN930
620 IF A$<>"S"THEN600
630 RUN
640 PRINT:PRINT:PRINT:PRINT
650 PRINT"ESCOGE NIVEL DE DIFICULTAD"
660 PRINT"ENTRE 10 Y 90"
670 INPUTXX%
680 IFXX%<10ORXX%>90 THEN PRINT"TTT";:GOTO650
690 XX%=100-XX%
700 RETURN
710 FORX=1TO20
720 PRINT"▀HUNDIDO!!▮▮▮▮▮▮▮▮";:FORYY=1TO50:NEXT
730 PRINT"▣HUNDIDO!!▮▮▮▮▮▮▮▮";:FORYY=1TO50:NEXT
740 NEXTX
750 PRINT:PRINT:PRINT"HAS NAVEGADO";
    MILES/1000;"KM":PRINT"A ";100-XX%;"KM/H"
760 PRINT
```

Juegos



Viene de la página anterior

```
770 PRINT:PRINT"OTRO JUEGO? (S/N)";
780 RETURN
790 LAST=(22+K)*5:XS=22+K:PO=KR+(PK-VR)
800 FORX=0TOLASTSTEPXS
810 POKE(PO+X),0:POKE(PK+X),81
820 NEXTX
830 POKE(PO+X-1),0:POKE(PO+X+1),0
840 POKE(PK+X-1),42:POKE(PK+X+1),42
850 POKE(PO+X-21),0:POKE(PO+X-22),0
860 POKE(PK+X-21),42:POKE(PK+X-22),42
870 POKE(PO+X-23),0:POKE(PO+X+21),0
880 POKE(PK+X-23),42:POKE(PK+X+21),42
890 POKE(PO+X+22),0:POKE(PO+X+23),0
900 POKE(PK+X+22),42:POKE(PK+X+23),42
910 PRINT:PRINT"ESTAS A PUNTO DE HUNDIRTE!!■"
920 GOTO590
930 PRINT"▣";:POKE36879,27:END
```



VIC - 20

La flecha astuta

Presentamos aquí un juego para dedos ágiles. Es más o menos rápido y requiere unos buenos reflejos, especialmente al principio, hasta que se aprenden de memoria las teclas que hay que pulsar.

En el juego interviene una flecha un tanto especial que, lanzada al aire por el ordenador, al comenzar la partida, puede ser dirigida por el jugador, haciéndola cambiar de dirección en pleno vuelo, mediante cuatro teclas del teclado.

El objetivo del juego es clavar la flecha en los bloques que van apareciendo aleatoriamente en la pantalla. Sólo dispone de unos segundos, desde que aparece el bloque, para clavar la flecha, pues al cabo de ese tiempo, el bloque desaparecerá. Cada bloque lleva un número en su interior, que son los puntos que consigue el jugador si acierta a clavar la flecha en él. Si la flecha se clava en los rebordes del rectángulo de juego, la partida habrá terminado. Además, la flecha, al moverse, va dejando tras de sí una estela multicolor, que también hay que evitar.

```
10 REM***LA FLECHA ASTUTA
20 REM***VIC-20[3,3K]
30 REM***ENERO 84
40 VR=PEEK(648)*256
50 KR=38400:IFVR<>7680THENKR=37888
60 GOSUB770
70 POKE828,0
80 DIMP(80)
90 KL(1)=1:KL(2)=3:KL(3)=5:KL(4)=7:S=0
100 FORI=1TO4:READA$(I):NEXT:DATAB,F,H,T
110 D(0)=22:D(1)=60:D(2)=62:D(3)=30:T6=3599
120 T9=VR
130 CS=VR
140 C1=22
144 REM
145 REM***REBORDE DE PANTALLA
146 REM
150 PRINT"■■ RECORD"PEEK(828)
155 FORP=1TO1000:NEXT
160 PRINT"■■■●●●●●●●●●●●●●●●●●●●●●";
170 FORI=1TO20
180 PRINT"●                    ●";:NEXT
190 PRINT"●●●●●●●●●●●●●●●●●●●●●●    ■■"
194 REM
195 REM***INICIO DE TIEMPO Y DEL JUEGO
196 REM
200 V=15:H=10:V1=0:H1=-1
210 P2=10:D1=1:TI$="000000"
220 PRINT"■";RIGHT$(TI$,2):IFTI>T6GOTO580
230 GETZ$:IFZ$=""THEN280
240 Z=50:FORY=1TO4:IFZ$=A$(Y)THENZ=Y-1:Y=5
250 NEXT:IFZ<>INT(Z)ORZ<0ORZ>3GOTO280
260 D1=Z:D=Z-1.5:V1=INT(ABS(D))*SGN(D)
270 H1=SGN(D)-V1
280 V=V-V1:H=H+H1:GOSUB740
290 P=CS+V*C1+H:KL=INT(RND(1)*4+1):KL=KL(KL)
300 POKEKR+V*C1+H,KL
310 P9=PEEK(P)
320 R6=R7:R7=R7+1:IFR7>P2THENR7=0
330 P1=P(R7):P(R7)=P
340 IFP1<>0THENPOKEP1,32:POKEP1-CS+KR,7
350 POKEP,D(D1):P1=P(R6):IFP1<>0THENPOKEP1,81
360 IFP9<>32GOTO490
370 IFRND(1)>.05THEN220
380 FORV3=V2-1TOV2+1:P3=V3*C1+T9
390 FORH3=H2-1TOH2+1:IFPEEK(P3+H3)<>102GOTO410
400 POKEP3+H3,32
410 NEXTH3,V3:T=0:POKEP8,32
420 V2=INT(RND(1)*18)+3:H2=INT(RND(1)*19)+2
430 FORV3=V2-1TOV2+1:P3=V3*C1+T9
440 FORH3=H2-1TOH2+1:IFPEEK(P3+H3)<>32GOTO420
450 NEXTH3,V3:FORV3=V2-1TOV2+1:P3=V3*C1+T9
460 FORH3=H2-1TOH2+1:POKEP3+H3,102
470 NEXTH3,V3:T=9*RND(1):P8=V2*C1+H2+T9
480 POKEP8,49+T:GOTO220
490 IFP9<>102THEN550
500 T$=TI$
```

Viene de la página anterior

```
510 T=T-1:S=S+1:POKEP8,T+49
520 PRINT"■■■■■■■■■■■";S
530 GOSUB690:IFT>=0THEN510
540 P2=P2+1:TI$=T$:GOTO380
544 REM
545 REM
546 REM
550 POKE36877,220:FORL=15TO0STEP-1:POKE36878,L
560 FORM=1TO300:NEXT:NEXT
570 POKE36877,0:POKE36878,0
580 IFS<=PEEK(828)THEN610
590 PRINT"■■■■■NUEVO RECORD"
600 PRINT"■"S"PUNTOS":POKE828,S:GOTO630
610 PRINT"■■■■■EL RECORD ES"; PEEK(828);
620 PRINT"■":PRINT"■■HAS OBTENIDO "S"■"
630 PRINT"■■■OTRO JUEGO (S/N) ?"
640 GETZ$:IFZ$=""THEN640
650 IFZ$="S"THEN RESTORE:PRINT"■":GOTO90
660 IFZ$<>"N"GOTO640
670 PRINT"■■";:POKE36879,27:END
680 REM MUSIC
690 POKE36878,15:FORL=200TO240:POKE36876,L
700 FORM=1TO5:NEXT:NEXT
710 POKE36878,0:POKE36876,0
720 :
730 RETURN
740 POKE36878,15:POKE36876,220
750 FORL=1TO5:NEXT:POKE36876,0
760 RETURN
764 REM
765 REM***INSTRUCCIONES
766 REM
770 POKE36879,42
780 PRINT"■■***LA FLECHA ASTUTA***"
790 PRINT"■MANEJA LA FLECHA"
800 PRINT"POR LA PANTALLA"
810 PRINT"SI ALCANZAS LAS DIANAS"
820 PRINT"OBTIENES LOS PUNTOS"
830 PRINT"QUE INDICAN"
840 PRINT"  T   T=ARRIBA"
850 PRINT" F    F=IZQUIERDA"
860 PRINT"   H  H=DERECHA"
870 PRINT"  B   B=ABAJO"
890 PRINT"■NO CHOQUES CON LAS"
900 PRINT"PAREDES NI CONTIGO"
905 PRINT"MISMO"
910 PRINT"LA FLECHA CRECE CON"
920 PRINT"EL PASO DEL TIEMPO"
930 PRINT"■TIENES 60 SEGUNDOS"
940 PRINT"■PULSA UNA TECLA■"
950 GETA$:IFA$=""THEN950
960 PRINT"■":RETURN
```

Diamantes

Todo el mundo sabe el valor que tienen los diamantes, por ello es seguro que, si de repente comenzaran a llover diamantes, usted saldría a la calle, con los recipientes que más a mano tuviera, a fin de recoger el mayor número posible de gotas de tan singular chaparrón. Con este juego, eso es exactamente lo que tiene que hacer.

En la parte superior de su pantalla aparecerán, al pasar el juego, seis filas de diamantes, y en la parte inferior seis bandejas. Los diamantes empezarán a caer, como si de lluvia se tratara, y usted, utilizando las teclas L y ; tendrá que procurar recogerlos sin que se le escape ninguno. Cada vez que recoja el último diamante de una fila, perderá una bandeja, de modo que para la última fila, sólo dispondrá de una de las seis bandejas iniciales. Sólo puede dejar escapar cinco diamantes, pues al sexto que pierda, termina el juego.

El programa dispone de varios niveles de dificultad, que podrá ir superando a medida que adquiere práctica. El máximo nivel de dificultad es para verdaderos expertos.

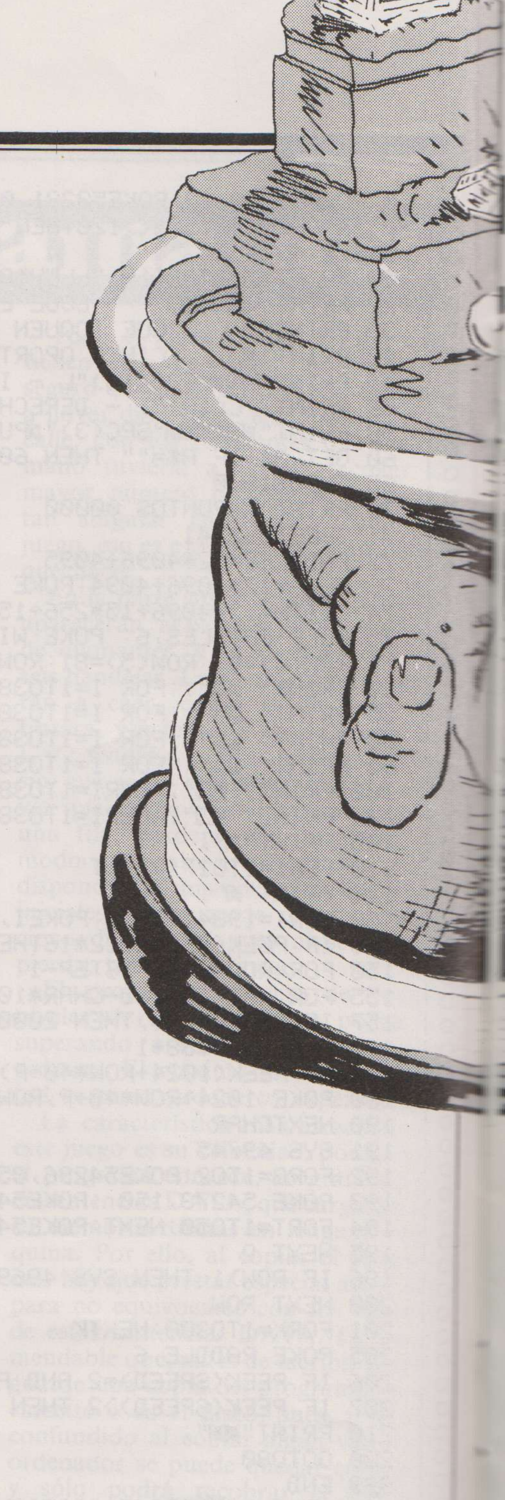
La característica más atractiva de este juego es su velocidad, por lo que el programa consiste, en gran parte, en sentencias DATA, que cargarán en su memoria rutinas en lenguaje máquina. Por ello, al copiar el programa, hay que prestar especial atención para no equivocarse con los valores de estas sentencias DATA. Es recomendable que, antes de escribir RUN, guarde una copia del programa en el cassette o en el disco, pues, si se ha confundido al copiar algún valor, el ordenador se puede quedar colgado, y sólo podrá recobrar el control apagándolo, con lo que perderá el programa almacenado en memoria. Si el programa no funciona, es casi seguro que tendrá algún número equivocado en alguna sentencia DATA, así que verifíquelas cuidadosamente.

COMMODORE 64

```
5 POKE53280,12:POKE53281,0
7 IF PEEK(49152)<>120THEN GOSUB49000
9 SYS 49745
10 PRINT"♥■"TAB(10)"LLUVIA DE DIAMANTES"
20 PRINT"QQQQQ■      COGE LOS DIAMANTES ANTES DE
30 PRINT"Q      QUE TOQUEN EL SUELO.TIENES
40 PRINT"Q      CINCO OPORTUNIDADES.
45 PRINT"QQ■"SPC(13)"L - IZQUIERDA
46 PRINTSPC(13)"; - DERECHA■"
50 PRINT"QQQQQ■"SPC(3)"■PULSA CUALQUIER TECLA PARA EMPEZAR"
60 GET A$:IF A$="" THEN 60
65 GOSUB1000
70 PRINT"♥■PUNTOS 00000    OPORTUNIDADES: ●●●● "
71 SPEED=53241
72 PADDLES=12*4096+4095
73 FLAG=12*4096+4094:POKE FLAG,0
74 WIDTH=12*4096+15*256+15*16+11
75 POKE PADDLES,6: POKE WIDTH,W: POKE SPEED,10-S
78 ROW(6)=81:ROW(5)=81:ROW(4)=207:ROW(3)=207:ROW(2)=90:ROW(1)=90
80 PRINT" ■■";:FOR I=1TO38:PRINT"♦";:NEXT:PRINT"■ ";
85 PRINT" ■■";:FOR I=1TO38:PRINT"♦";:NEXT:PRINT"■ ";
90 PRINT" ■■";:FOR I=1TO38:PRINT"┐";:NEXT:PRINT"■ ";
95 PRINT" ■■";:FOR I=1TO38:PRINT"┐";:NEXT:PRINT"■ ";
100 PRINT" ■■";:FORI=1TO38:PRINT"o";:NEXT:PRINT" ";
102 PRINT" ■■";:FORI=1TO38:PRINT"o";:NEXT:PRINT" ";
105 PRINT"■";
110 FORI=1TO17:PRINT"                                        ";:NEXT
120 PRINT"■";
130 FORI=1984TO2023:POKEI,248:POKEI+54272,10:NEXT
140 IF PEEK(789)<>12*16THENSYS 12*4096
150 FOR ROW=6 TO 1STEP-1: FOR CHAR=1TO38
155 FOR K=1 TO 600-CHAR*10+(6-ROW)*20-50*(9-PEEK(SPEED)):NEXT
157 IF PEEK(FLAG)THEN 2000
160 P=RND(1)*38+1
170 IF PEEK(1024+ROW*40+P)=32THEN160
180 POKE 1024+ROW*40+P,ROW(ROW)
190 NEXTCHAR
191 SYS 49745
192 FORQ=1TO2:POKE54296,05:POKE54277,5:POKE54278, 218
193 POKE 54273,150 :POKE54272,139:POKE54276,17
194 FORT=1TO50:NEXT:POKE54276,16:FORT=1TO10:NEXT
195 NEXT Q
196 IF ROW>1 THEN SYS 49691
200 NEXT ROW
201 FORK=1TO300:NEXTK
205 POKE PADDLE,6
206 IF PEEK(SPEED)=2 AND PEEK(WIDTH)>1 THEN POKE WIDTH,PEEK(WIDTH)-1
207 IF PEEK(SPEED)>2 THEN POKE SPEED,PEEK(SPEED)-1
210 PRINT"■Q";
220 GOTO80
999 END
1000 PRINT"♥"SPC(7)"DIFICULTAD    QQQQQ"
1010 INPUT"■VELOCIDAD (1-9)■■■■5■■■";S
1015 IF S>9 OR S<1 THEN 1010
1020 INPUT"QQQ■ANCHURA DE CESTAS (1-9)■■■■4■■■";W
1030 IF W>9 OR W<1 THEN1020
1040 RETURN
2000 PRINT"QQQQQQQQQQQ■-------------------FIN-------------------";
```

Viene de la página anterior

```
 2001 PRINT"╥PULSA BARRA-ESPACIOS PARA JUGAR DE NUEVO"
 2010 POKE198,0
 2020 GETA$:IFA$<>" "THEN 2020
 2030 RUN65
49000 PRINT"▬⊏▩▩CARGANDO LENGUAJE MAQUINA...▩▩▩":
      TI$="000000"
49005 I=49152
49007 PRINT"LISTO EN"STR$(29-VAL(TI$))" SEGUNDOS ⅂"
49010 READ A:IF A=256 THEN RETURN
49020 POKE I,A:I=I+1:GOTO 49007
49152 DATA 120,169,192,141,21,3,169
49160 DATA 29,141,20,3,88,169,18
49168 DATA 141,253,207,169,0,141,250
49176 DATA 207,141,247,207,141,248,207
49184 DATA 96,173,255,207,141,252,207
49192 DATA 172,253,207,169,32,153,151
49200 DATA 7,200,169,160,174,251,207
49208 DATA 153,151,7,200,202,208,249
49216 DATA 169,32,153,151,7,206,252
49224 DATA 207,208,3,76,3,193,172
49232 DATA 253,207,169,32,153,71,7
49240 DATA 200,169,160,174,251,207,153
49248 DATA 71,7,200,202,208,249,169
49256 DATA 32,153,71,7,200,206,252
49264 DATA 207,208,3,76,3,193,172
49272 DATA 253,207,169,32,153,247,6
49280 DATA 200,169,160,174,251,207,153
49288 DATA 247,6,200,202,208,249,169
49296 DATA 32,153,247,6,200,206,252
49304 DATA 207,240,123,172,253,207,169
49312 DATA 32,153,167,6,200,169,160
49320 DATA 174,251,207,153,167,6,200
49328 DATA 202,208,249,169,32,153,167
49336 DATA 6,200,206,252,207,240,91
49344 DATA 172,253,207,169,32,153,87
49352 DATA 6,200,169,160,174,251,207
49360 DATA 153,87,6,200,202,208,249
49368 DATA 169,32,153,87,6,200,206
49376 DATA 252,207,240,59,172,253,207
49384 DATA 169,32,153,7,6,200,169
49392 DATA 160,174,251,207,153,7,6
49400 DATA 200,202,208,249,169,32,153
49408 DATA 7,6,200,206,252,207,240
49416 DATA 27,172,253,207,169,32,153
49424 DATA 183,5,200,169,160,174,251
49432 DATA 207,153,183,5,200,202,208
49440 DATA 249,169,32,153,183,5,200
49448 DATA 165,197,201,42,208,13,173
49456 DATA 253,207,201,1,240,24,206
49464 DATA 253,207,76,40,193,201,50
49472 DATA 208,14,173,253,207,24,109
49480 DATA 251,207,201,39,240,3,238
49488 DATA 253,207,238,250,207,173,250
49496 DATA 207,205,249,207,240,3,76
49504 DATA 49,234,169,0,141,250,207
49512 DATA 169,112,133,251,169,7,133
```






```
49520 DATA 252,160,0,185,152,7,41
49528 DATA 127,201,32,208,74,200,192
49536 DATA 39,208,242,160,0,177,251
49544 DATA 201,81,240,37,201,207,240
49552 DATA 33,201,90,240,29,200,192
49560 DATA 40,208,237,56,165,251,233
49568 DATA 40,133,251,176,2,198,252
49576 DATA 166,251,208,220,166,252,224
49584 DATA 4,208,214,76,49,234,170
49592 DATA 152,24,105,40,168,138,145
49600 DATA 251,152,56,233,40,168,169
49608 DATA 32,145,251,32,251,193,76
49616 DATA 99,193,169,32,153,152,7
49624 DATA 32,81,194,169,15,141,24
49632 DATA 212,169,17,141,5,212,169
49640 DATA 213,141,6,212,169,2,141
49648 DATA 3,212,169,100,141,2,212
49656 DATA 169,5,141,1,212,169,135
49664 DATA 141,0,212,169,65,141,4
49672 DATA 212,160,0,162,0,142,32
49680 DATA 208,232,208,250,200,208,247
49688 DATA 169,12,141,32,208,169,64
49696 DATA 141,4,212,160,39,185,0
49704 DATA 4,201,81,240,11,136,208
49712 DATA 246,169,1,141,254,207,76
49720 DATA 49,234,169,32,153,0,4
49728 DATA 76,49,234,152,72,160,10
49736 DATA 185,0,4,201,57,208,9
49744 DATA 169,48,153,0,4,136,76
49752 DATA 255,193,185,0,4,24,105
49760 DATA 1,153,0,4,104,168,96
49768 DATA 174,255,207,202,142,255,207
49776 DATA 232,169,152,133,251,169,7
49784 DATA 133,252,56,165,251,233,80
49792 DATA 133,251,176,2,198,252,202
49800 DATA 208,242,160,0,177,251,201
49808 DATA 160,240,4,200,76,59,194
49816 DATA 174,251,207,169,32,145,251
49824 DATA 200,202,208,250,96,160,0
49832 DATA 152,153,0,212,200,192,9
49840 DATA 208,248,96,256
```

Juegos

Si le gustan las situaciones arriesgadas, atrévase con este juego con el que, en cualquier momento, puede saltar por los aires.
En su pantalla aparecerá una cuadrícula.
El ordenador ha colocado minas en algunos de los cuadrados de la cuadrícula, de forma aleatoria, pero usted no sabe en cuáles.




```
1 PRINT"♥"
2 REM**CAMPO DE MINAS
5 POKE36879,42
8 GOSUB2000
10 DIMM(10,11)
20 L=L+1:J=441
27 PRINT"♥■"
28 FORI=1TO10
30 PRINT" - - - - - - - - - -"
40 PRINT"| | | | | | | | | | |"
50 NEXT
53 PRINT" - - - - - - - - - -"
55 IFEOG=1THEN RETURN
56 POKE7680+J,102:POKE7680+61,42:X=2:Y=2
57 FORI=1TO4+L*3:A=INT(RND(.5)*9):B=INT(RND(.5)*10):M(A,B)=1
58 IF(A=0ANDB=0)OR(A=9ANDB=10)OR(A=1ANDB=0)THENM(A,B)=0
59 NEXT
60 IFX=10ANDY=11THEN5000
61 GOSUB1500
62 Q=J
64 GETP$:IFP$=""THEN60
70 IFP$="I"THEN1000
75 IFP$="M"THEN1200
80 IFP$="A"THEN1400
85 IFP$="D"THEN1600
90 GOTO64
1000 IFJ<66THEN60
1020 J=J-44:POKE7680+Q,46:POKE7680+J,102
1022 Y=Y+1
1025 GOSUB 1500
1040 GOTO60
1200 IFJ>412THEN60
1220 J=J+44:POKE7680+Q,46:POKE7680+J,102
1222 Y=Y-1
1225 GOSUB1500
1240 GOTO 60
1400 IF INT((J-1)/22)=(J-1)/22THEN 60
1420 J=J-2:POKE7680+Q,46:POKE7680+J,102
1422 X=X-1
1425 GOSUB 1500
1440 GOTO 60
1500 REM BOMBS
1502 IFM(X-2,Y-2)=1 THEN3000
1510 X1=X+1
1511 IF M(X1-2,Y-1)=1THEN B1=B1+1
1512 IFM(X1-2,Y-2)=1THEN B1=B1+1
1513 IFY<3THEN1515
1514 IFM(X1-2,Y-3)=1THEN B1=B1+1
1515 X1=X
1516 IFM(X1-2,Y-1)=1  THEN B1=B1+1
1517 IFM(X1-2,Y-2)=1  THENB1=B1+1
1518 IFY<3THEN1520
1519 IFM(X1-2,Y-3)=1  THENB1=B1+1
1520 X1=X-1:IFX<3THEN 1570
1521 IFM(X1-2,Y-1)=1 THEN B1=B1+1
1522 IFM(X1-2,Y-2)=1 THEN B1=B1+1
1523 IFY<3THEN1570
1524 IFM(X1-2,Y-3)=1  THEN B1=B1+1
1570 PRINT"■"B1"BOMBA(S)A UN CUADRADO"
1575 B1=0
1599 RETURN
```

```
1600 IF INT((J+5)/22)=(J+5)/22 THEN60
1620 J=J+2:POKE7680+Q,46:POKE7680+J,102
1622 X=X+1
1625 GOSUB1500
1640 GOTO60
1998 GETZ$: IFZ$=""THEN1998
1999 END
2000 PRINT"BIENVENIDO, CAMPO DE  MINAS"
2001 PRINT"ENCUENTRE UN CAMINO SEGURO"
2002 PRINT"MISION: "
2003 PRINT"ALCANZAR LA ESQUINA   SUPERIOR DERECHA"
2004 PRINT"EL RADAR MUESTRA SOLO MINAS DISTANTES "
2005 PRINT"UN CUADRADO"
2006 PRINT"A=IZQD. D=DER. I=ARR. M=ABAJ."
2008 PRINT"BARRA DE ESPACIOS":PRINT"PARA COMENZAR"
2009 GETSW$:IFSW$=" "THEN PRINT"":RETURN
2010 GOTO 2009
3000 PRINT"DESTRUIDO!"
3002 PRINT"↑ PARA VER BOMBAS"
3004 GETSP$:IFSP$="↑"THEN3008
3006 GOTO 3004
3008 EOG=1:GOSUB27
3009 POKE7680+J,102
3010 FORA=0TO10:FORB=0TO11
3015 IFM(A,B)=1 THENSS=A*2+1+(9-B)*44+44:POKE 7680+SS,81
3020 NEXTB,A
3040 PRINT"OTRO JUEGO (S/N) ?"
3045 GETG$:IFG$="S"THENRUN
3050 IFG$="N"THENPRINT"ADIOS!":END
3055 GOTO 3045
5000 REM
5002 PRINT"TUVISTE SUERTE EN EL  NIVEL ";L
5005 FORI=0TO10:FORK=0TO11:M(I,K)=0:NEXTK,I
5010 PRINT"SI PRESIONAS ↑ TE       PONDRE EN EL NIVEL ";L+1
5020 PRINT"ON LEVEL ";L+1
5030 GETSP$:IFSP$="↑"THEN PRINT"":GOTO20
5040 GOTO 5030
```

Su misión será alcanzar el cuadrado superior derecho, para lo cual debe moverse por la pantalla mediante las teclas que el programa le señalará.

Sólo cuenta para guiarse, con un rudimentario radar, que le dice en todo momento cuantas minas hay en los cuadrados colindantes con el suyo.

Si tiene la mala suerte de pisar una mina, su ordenador le dirá tranquilamente que ha sido destruido y señalará donde había colocado todas y cada una de las minas.

Si por el contrario alcanza su objetivo sano y salvo no cante victoria; aparecerá un nuevo campo con más minas que el anterior. ¡Aquí el que no corre, vuela!





Uso de las teclas de función

Las cuatro teclas de función del **VIC-20** no corresponden a ningún comando específico, sino que están pensadas para que, cada usuario, las utilice según sus necesidades, y desde este punto de vista, lo que con ellas se puede hacer, sólo está limitado por la habilidad del programador.

Es curioso, sin embargo, que el manual suministrado con el ordenador, ignore casi por completo el manejo y operación de estas cuatro teclas.

Aunque pueden escribirse rutinas para asignar comandos específicos a estas teclas, el uso más corriente de las mismas es a través del programa. Esto se realiza mediante la instrucción GET A$. Cuando pulsamos una de las teclas, estamos colocando, en un registro del teclado, un valor numérico que es el valor de la tecla en código ASCII. Las teclas de función dan lugar, según este código, a valores comprendidos entre 133 y 140. Para hacer uso del valor numérico del código ASCII, se emplea la función CHR$(n), que realiza la operación inversa, es decir, determina qué tecla tiene, como valor en el código, el número n.

Reuniendo estos elementos, el programa puede saber qué tecla hemos pulsado, y entonces le podemos decir que si se pulsa una tecla de función haga esto o lo otro.

Para ilustrar todo esto, damos el siguiente programa, que cuando se pulsa la tecla f1 escribirá "F1 PULSADA" y así con las cuatro teclas. En lugar de poner PRINT"F1 PULSADA"; podemos poner cualquier cosa que se nos ocurra; con la seguridad de que el programa hará lo que le decimos, cuando se pulse la tecla f1.

VIC - 20

```
10 REM***USO DE TECLAS DE FUNCION***
20 GETA$:IFA$=""GOTO20
30 IFA$=CHR$(133)THEN PRINT"F1 PULSADA":GOTO20
40 IFA$= CHR$(137)THENPRINT"F2 PULSADA":GOTO20
50 IFA$=CHR$(134)THENPRINT"F3 PULSADA":GOTO20
60 IFA$=CHR$(138)THENPRINT"F4 PULSADA":GOTO20
70 IFA$=CHR$(135)THENPRINT"F5 PULSADA":GOTO20
80 IFA$=CHR$(139)THENPRINT"F6 PULSADA":GOTO20
90 IFA$=CHR$(136)THENPRINT"F7 PULSADA":GOTO20
100 IFA$=CHR$(140)THENPRINT"F8 PULSADA":GOTO20
110 GOTO20
```

Salta ranas

Existen numerosos juegos de ordenador basados en pasatiempos de tipo matemático, como, "a que no consigues pasar de aquí a aquí en tres movimientos" o algo por el estilo, y que muchas veces nos propone algún amigo, tomando un café, y sacando unas monedas o algunas cerillas, que coloca de cierta manera delante de nosotros mientras nos dice: "a que no consigues..." Este juego es de ese tipo, aunque en lugar de utilizar monedas, se usan ranas, principalmente para poder saltar de una posición a otra con facilidad.

En la pantalla aparecen diez ranas colocadas en fila, cinco verdes a la izquierda y cinco blancas a la derecha, con un espacio en blanco entre los dos bandos. El objetivo del juego es colocar las ranas verdes donde están las blancas, y claro, que las ranas blancas pasen a ocupar la posición de las verdes, y ello con el mínimo número de movimientos. Para mover las ranas se emplean las teclas numéricas, teniendo presente que sólo hay dos movimientos permitidos; que la rana salte al espacio vacío cuando lo tiene al lado, o que salte al espacio vacío por encima de otra rana, pero sólo por encima de una.

El programa va contando los movimientos que se llevan a cabo y cuando termina el juego le dice el número total de los mismos y comenta su puntuación con frases como, ¡no está mal!, o, ¡excelente!, o algo así. Es a causa de estas frases, por la cantidad de memoria que utilizan, por lo que hay que añadir el módulo de expansión de 3K.

Existe un límite mínimo de movimientos que, seguro que será capaz de alcanzar después de jugar unas cuantas partidas.

VIC - 20 CON EXPANSION DE 3K.

```
0 REM***SALTA-RANAS***
10 POKE56,PEEK(56)-2:POKE52,PEEK(56)-2:POKE51,
   PEEK(55):CLR
20 PRINT"█":POKE36879,42
30 GOSUB2000
40 POKE36869,255
100 DIMA(19)
110 F$(0)=" "
120 F$(1)="█D"
130 F$(2)="█D"
150 AN$(1)="█?"
160 AN$(2)="█?"
170 AN$(3)=F$(0)
200 PP$="██████"
220 REM
260 T1$="███  ▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀█"
270 T1$=T1$+CHR$(13)+"    "
280 T1$=T1$+"█ *SALTA-RANAS* █"
520 FORK=1TO5
530 A(K)=1:A(K+6)=2
540 NEXT:A(6)=0
550 C=0
560 PRINTT1$:PRINT:PRINT
570 FORK=1TO11
580 PRINTF$(A(K));
590 NEXT:PRINT"██"
600 PRINT
610 PRINT"███123456789:;"
620 PRINTPP$"█████"
630 PRINT"██TU MUEVES              - █";
640 GOSUB1340:FF=0
650 PRINT"█":TD=1500
660 IFA(S)=0THEN1470
670 IFABS(S-E)>2THEN1500
690 IFE>11THENFF=1:E=12
700 X=0
710 PRINTPP$;TAB(S-1);
720 IFA(S)=1 THEN PRINTAN$(1);:GOTO750
730 PRINTAN$(2);
750 FORL=1TO500:NEXT
760 PRINT"█"AN$(3);
770 FORL=1TO50:NEXT
780 PRINTPP$;TAB(E-1);AN$(A(S));
790 FORL=1TO300:NEXTL
800 FORK=1TO150:NEXT
810 PRINT"█"F$(A(S));
820 C=C+1
830 A(E)=A(S):A(S)=0
840 FORI=1TO6
850 X=X+A(I)*10↑I
860 NEXT
870 IFFFTHEN1600
880 IF INT(X)<>222220THEN620
940 POKE36869,240:PRINT"██████":IFC<60THEN990
950 POKE36869,240:PRINT"██▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
960 PRINT"██ACABASTE POR FIN!!!          "
970 GOTO1020
990 IFC<50THEN1060
1000 PRINT"██▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀▀"
1010 PRINT"██NO ESTA MAL!!!      "
```

Juegos

```
1020 PRINT"■REALIZASTE"C"MOVIMIENTOS          "
1030 PRINT"PUEDES HACERLO MEJOR!!"
1040 GOTO1240
1060 IFC<40THEN1120
1070 PRINT"■■▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁"
1080 PRINT"■MUY BIEN!!!"
1090 PRINT"■HAS CONSEGUIDO    TERMINAR EL JUEGO"
1100 PRINT"SOLO"C"MOVIMIENTOS Y POR ENCIMA DE LA
     MEDIA":GOTO1240
1120 IFC=35 THEN 1190
1130 PRINT"■■▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁"
1140 PRINT"■■EXCELENTE!! ERES UN EXPERTO!!"
1150 PRINT"■LO HAS HECHO EN"C"MOVIMIENTOS"
1160 PRINT"ES CASI EL MEJOR RESULTADO POSIBLE"
1170 GOTO1240
1190 PRINT"■■▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁"
1200 PRINT"■■FELICIDADES  NADIE
     PUEDE HACERLO MEJOR"
1210 PRINT"■HAS COMPLETADO EL JUEGO EN 35 PASOS"
1220 PRINT"ESTE ES EL MINIMO"
1240 PRINT"■■JUEGAS OTRA VEZ?(S/N)"
1250 GETA$:IFA$<>"S"ANDA$<>"N"THEN1250
1260 IFA$="S"THEN POKE36869,255:GOTO520
1261 POKE56,PEEK(56)+2
1262 POKE 36869,240
1263 POKE 36879,27
1270 PRINT"■■GRACIAS POR JUGAR"
1280 END
1340 POKE198,0
1350 DEL=0:PRINT"■"
1360 PRINTTAB(23);"■DE                ■■";
1370 PRINTTAB(28);"■ ■■";
1380 GOSUB1820:IF DEL THEN PRINT"■";:GOTO1350
1390 S=VAL(C$):PRINT"■ A     ■■";
1400 FORI=1TO11:IFA(I)=0THENC$=STR$(I)
1401 NEXT
1402 PRINT"■■"C$;
1410 E=VAL(C$):RETURN
1470 PRINT"■■NO HAY NADIE EN LA  POSICION #"S"■.■"
1480 PRINT"■JUEGE OTRA VEZ":GOTO1560
1500 PRINT"■■OYE,NO PUEDO SALTAR    TAN LEJOS"
1560 TD=3000
1570 FORK=1TOTD:NEXT
1580 TD=3000:GOTO560
1600 PRINTPP$;TAB(11);F$(0);"■";
1610 PRINTF$(A(E));
1620 A$="%##%!!■■■■■■■"
1630 PRINT"■■■■";TAB(12);
1640 FORK=1TO20
1650 PRINT"■"A$;:A=SIN(3)
1660 PRINT"■"A$;:A=SIN(3)
1670 NEXT:PRINT"        ■■■■■"
1680 PRINT"▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁▁"
1690 PRINT"■MIRA LO QUE HAS HECHO!"
1700 PRINTPP$;"■";TAB(34);AN$(0);
1710 FORK=72TO0STEP-4
1720 PRINTF$(A(E));"■";AN$(0);
```



```
1730 FORL=1TOK:NEXT
1740 PRINTF$(0)+"▮▮";
1750 NEXT:A(12)=0
1760 PRINTF$(0);"IIIII":GOTO1240
1820 GETC$:IFC$=""THEN1820
1830 IFC$=CHR$(20)THENDEL=1:RETURN
1840 IFC$<"1"ORC$>"9"THEN1820
1850 PRINT"▮"C$;:IFC$<>"1"THEN RETURN
1860 IFA(8)*A(9)*A(10)*A(11)THENRETURN
1870 PRINT"▮ ▮▮";
1880 GETCC$:IFCC$=""THEN1880
1890 IFCC$=CHR$(20)THENDEL=1:RETURN
1900 IFCC$<"0"ORCC$>"1"THEN1880
1910 C$=C$+CC$:PRINT"▮"CC$;:RETURN
2000 FORI=1TO15
2010 READX
2020 FORJ=0TO7
2030 READK
2040 POKEX+J,K
2050 NEXTJ,I
2060 RETURN
2100 DATA 7424,0,0,0,0,0,0,0,0
2101 DATA 7560,16,48,16,16,56,0,0,0
2102 DATA 7568,56,8,56,32,56,0,0,0
2103 DATA 7576,56,8,56,8,56,0,0,0
2104 DATA 7584,40,40,60,8,8,0,0,0
2105 DATA 7592,56,32,56,8,56,0,0,0
2106 DATA 7600,56,32,56,40,56,0,0,0
2107 DATA 7608,56,8,16,16,16,0,0,0
2108 DATA 7616,56,40,56,40,56,0,0,0
2109 DATA 7624,56,40,56,8,56,0,0,0
2110 DATA 7632,46,106,42,42,126,0,0,0
2111 DATA 7640,36,108,36,36,126,0,0,0
2112 DATA 7664,24,189,255,60,126,66,195,0
2113 DATA 7672,153,189,126,60,126,66,66,195
2114 DATA 7552,56,40,40,40,56,0,0,0
```



COMMODORE 64

El Commodore 64 dispone de la interesante función TI$, que la mayoría de usuarios conocerá, y que permite acceder al reloj interno de la máquina para su empleo en programas. Sin embargo, el empleo de esta función, está limitado a intervalos de horas, minutos y segundos. A veces, puede ser interesante disponer de décimas de segundo.

Pues bien, la rutina que presentamos va más lejos todavía, permitiendo contar hasta en centésimas de segundo. El secreto está en que la función TI lee el reloj interno cada 1/60 segundos. Entre dos de estas lecturas, se introduce una instrucción de duración determinada y, posteriormente, unas sencillas operaciones con las dos lecturas, nos proporcionan el intervalo de centésimas de segundo.

```
1000 PRINT"▮"
1020 T1=TI
1030 GETA$
1040 T2=TI
1050 T=INT(((T2-T1/60)*100+.5)/100)
1060 PRINT"▮"T:PRINT"▮▮▮▮▮▮▮▮SEGS"
1070 IFA$=""GOTO1030
```

Lenguaje máquina:

Microproce

Es normal que una vez pasado el período de aprendizaje en el manejo de ordenador, deseemos ampliar nuestros conocimientos. Es corriente también oír hablar sobre el lenguaje máquina, asunto que inmediatamente asociamos con algo muy complejo. Nada más lejos de la verdad, utilizar el lenguaje máquina es difícil si no sabemos qué se puede hacer con él, cómo y para qué sirve.

El conocimiento de los microprocesadores se revela fundamental para conseguir un mayor aprovechamiento de las posibilidades del ordenador. Con esta serie de artículos pretendemos descifrar los misterios de la familia 6500, utilizada por Commodore.

La configuración más elemental de un sistema basado en microprocesador es la de la figura 1.

En este diagrama de bloques se pueden distinguir los siguientes módulos: CPU (Unidad Central de Proceso) cuya misión principal y prioritaria consiste en interpretar y ejecutar las instrucciones recibidas desde la memoria.

El bus de datos, que en este caso es de 8 bits, es el encargado de transferir todos los datos del sistema entre la CPU, la memoria y los módulos de entrada y salida.

El bus de direcciones está compuesto de 16 líneas de carácter unidireccional, la CPU pone en él la dirección de memoria, RAM, ROM, o de Entrada/Salida a la que desea acceder. Los módulos de memoria de programa (ROM), memoria de datos (RAM) y módulos de entrada/salida, completan el sistema.

LA CPU Como hemos indicado anteriormente, la misión fundamental de la unidad central de proceso es interpretar y ejecutar las instrucciones recibidas, procedentes de la memoria, de tal forma que el código binario de la instrucción que está en curso, se recibe por el bus de datos (de 8 bits en este microprocesador) como se ve en el diagrama de la figura 2.

Decodificador de instrucciones: Se encarga de seleccionar (decodificar) las posiciones que corresponden a una instrucción en la memoria ROM, donde están grabados los códigos de las operaciones más elementales que componen las instrucciones.

Secuenciador: Es el encargado de generar señales que gobiernan todos los elementos del sistema que se necesitan para llavar a buen término una instrucción.

Contador de programa: Este módulo es el encargado de enviar, a través del bus de direcciones, la posición de memoria correspondiente donde se encuentra la próxima instrucción a realizar. El contador de programa se incrementa en una unidad en el momento en que la memoria acepta la dirección anterior. En otras situaciones en contenido se altera de forma radical. En otro capítulo veremos cómo.

La ALU: Unidad aritmética lógica, es la unidad que se encarga de ejecu-

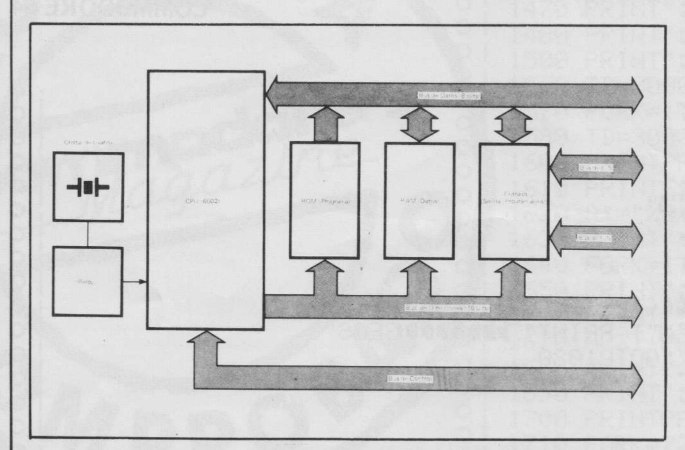
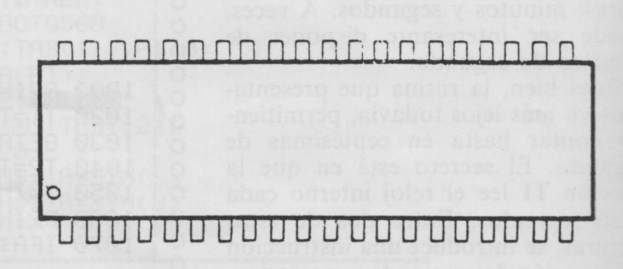


Figura 1. Diagrama de bloques clásico de un microordenador.


Figura 2. Aspecto físico del 6502 visto desde arriba.

esadores

Familia 6500

tar todas las operaciones aritméticas/lógicas de transferencia.

Acumulador: Es uno de los registros memoria) más importantes de la CPU. Es un registro de 8 bits, que tiene la misión de recibir siempre uno de los operandos que participan en la operación. Además de este registro, es donde se deposita el resultado final de la operación. Como podemos apreciar en el diagrama de bloques, la ALU recibe uno de los operandos del acumulador, y el otro operando de otro registro auxiliar, también de 8 bits.

Todas las operaciones realizadas en la ALU, afectan al registro de estado.

Registro de estado: Es un registro, también de 8 bits, actuando todos como alarmas (flags). Con estas informaciones podremos saber si la paridad es par o impar, si el signo es negativo, si ha habido acarreo o desbordamiento (overflow), etcétera, al ejecutar una instrucción.

Registro índice: El contenido de este registro se suele utilizar con el fin de poder localizar ciertas posiciones de la memoria.

Registros de trabajo: Estos registros se utilizan internamente para manipular datos de interés en la resolución de programas. No todos los microprocesadores incorporan este tipo de registros.

Registro Stack Pointer: Este registro es un contador que controla una zona de la memoria, en la cual se depositan de una forma temporal, los contenidos de ciertas posiciones de memoria, que más tarde serán utilizados.

LA CPU 6502

Es una de las CPU más utilizadas. En realidad el microordenador Commodore 64 utiliza una CPU 6510, que es idéntica internamente a la 6502 y utiliza las mismas instrucciones, la única diferencia es que la 6510 tiene un port de I/0 de 8 bits adicionales para gestión de memoria. Esto permite al microprocesador acceder a una memoria de tamaño superior a los 84

Figura 3. Generación de señales de reloj sin cuarzo.

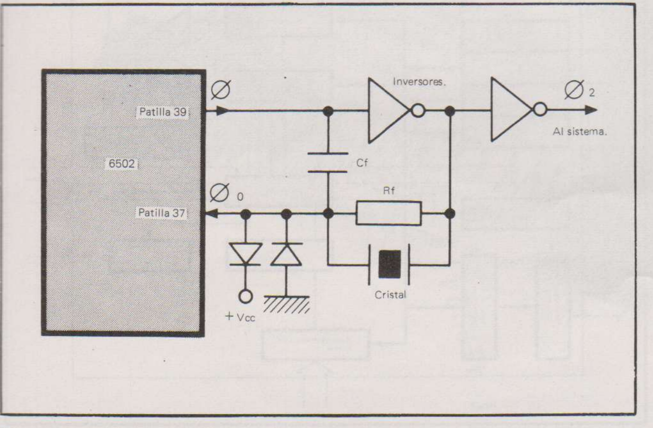


Figura 4. Señales de reloj mediante cuarzo.

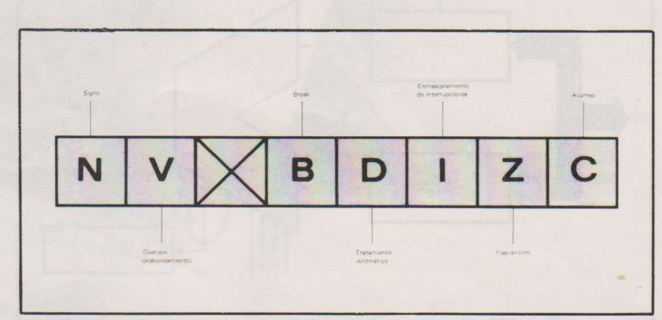


Figura 5. Los 8 bits del Registro de Estado.

Familia 6500

Kbytes. Con 16 bits sólo se pueden direccionar hasta 64 Kbytes.

La CPU 6502, es una pastilla de 40 patillas cuyas características más importantes son:

— Alimentación única de +5 V.
— Procesamiento paralelo de palabras de 8 bits.
— Juego de 56 instrucciones.
— Trece modos de direccionamiento distintos.
— Aritmética decimal y binaria.
— Stack Pointer de 8 bits programable.
— Generador interno de reloj.
— Interrupciones mascarables y no enmascarables.
— Capacidad de direccionamiento de 64 K (6502) y 84 K (6510).
— Frecuencia de trabajo entre 1 y 2 MHz (Ciclo de máquina de 1 msg. o 500 nsg.).
— La instrucción más rápida dura dos ciclos de máquina y la más lenta es de 7 ciclos de máquina.
— Puede usarse con cualquier tipo de memorias.

CONEXIONADO DE LAS PATILLAS

Patilla 1 (vss): Se conecta a masa (Ov).

Patilla 2 (RDY): Adapta la CPU a la velocidad de las memorias. La señal RDY detiene a la CPU en todos los ciclos menos en los de escritura, dejando en estado flotante los buses de direcciones y datos.

Patilla 3 (ϕ1 out): Salida de la señal de sincronismo de su reloj interno.

Patilla 4 (IRQ): Señal de interrupción desde el exterior, activa con nivel bajo.

Patilla 5 (NC): Esta patilla no es necesario conectarla.

Patilla 6 (NMI): Señal de interrupción desde el exterior, activa con nivel bajo.

Patilla 7 (SYNC): Línea de salida de la CPU que sirve para identificar los ciclos en que se realiza la búsqueda de un código de operación.

Patilla 8 (Vcc): Patilla de alimentación a +5v de cc (corriente continua).

Patilla 9 a patilla 20 y patilla 22 a 25 (AB0 - AB15): Son las 16 líneas del bus de direcciones que, como ya dijimos anteriormente, es unidireccional. Estas líneas funcionan a nivel TTL (Ov o 5v).

Patilla 21 (Vss): Esta patilla junto con la patilla 1 se conecta a masa.

Patilla 26 a patilla 33 (DB7 - DB0) Son las ocho líneas que conforman el bus de datos y el bus de instrucciones que, como sabemos, es un bus bidireccional) y triestado (Ov,+5v o desconectado).

Patilla 34 (R/W): Línea de lectura y escritura, que sólo pasa a nivel bajo cuando se realiza una escritura de datos correctamente en la memoria o en periféricos exteriores.

Patillas 35 y 36 (Nc): No tienen conexión alguna.

Patilla 37 (ϕo IN): Señal de reloj exterior para estabilizar correctamente la frecuencia. Esta señal puede generarse gracias a un cristal de cuarzo o mediante una red R-C (resistencia-condensador) según las figuras 6 y 7.

Patilla 38 (S. O.): Señal exterior a nivel TTL para activación del flag de desbordamiento.

Patilla 39 (ϕ2 out): Tiene la misma función que la patilla 3.

Patilla 40 (RES): Señal exterior de interrupción, activa a nivel bajo. (Ver señales de interrupción en el capítulo siguiente).

Miguel Angel de Frutos

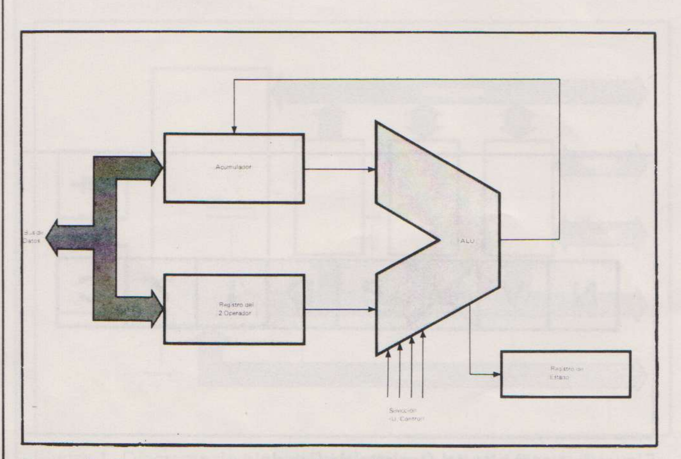


Figura 6. Toda operación en la ALU afecta al registro de estado.

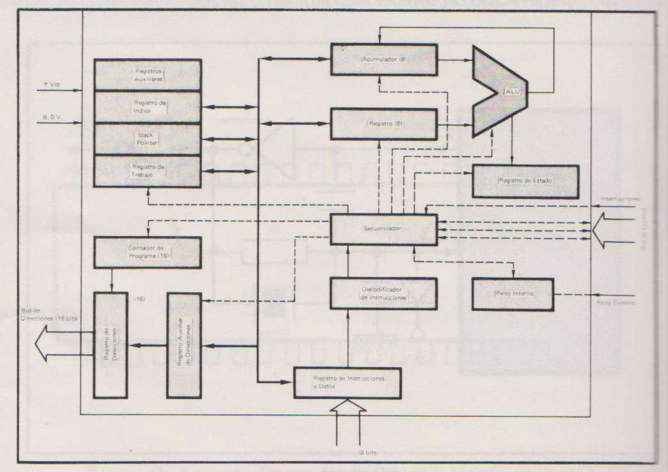


Figura 7. Diagrama interno del microprocesador

SU PROGRAMA PARA CUALQUIER SISTEMA COMMODORE PUEDE HACERLE GANAR 5.000 PTAS.

EL PRESENTE CONCURSO ESTA ABIERTO A TODOS NUESTROS LECTORES Y SU PARTICIPACION E INSCRIPCION ES GRATUITA.
LEA LAS BASES DEL CONCURSO

- NO SE ESTABLECEN LIMITACIONES EN CUANTO A EXTENSION, TEMA ELEGIDO O MODELO DE ORDENADOR
- LOS CONCURSANTES DEBERAN ENVIARNOS A LA DIRECCION QUE FIGURA AL PIE, EL CASSETTE O DISKETTE CONTENIENDO EL PROGRAMA, UNA EXPLICACION DEL MISMO Y, AL SER POSIBLE, UN LISTADO EN PAPEL DE IMPRESORA. SE PODRAN ENVIAR TANTOS PROGRAMAS COMO SE DESEE
- LOS PROGRAMAS, PREVIA SELECCION, SERAN PUBLICADOS EN LA REVISTA, OBTENIENDO TODOS ELLOS 5.000 PTAS.
- LA DECISION SOBRE LA PUBLICACION O NO DE UN PROGRAMA CORRESPONDE UNICAMENTE AL JURADO NOMBRADO AL EFECTO POR "COMMODORE MAGAZINE", SIENDO SU FALLO INAPELABLE
- LOS CRITERIOS DE SELECCION SE BASARAN EN LA CREATIVIDAD DEL TEMA ELEGIDO Y LA ORIGINALIDAD Y/O SENCILLEZ EN EL METODO DE PROGRAMACION GLOBAL
- ENVIAR A:
CONCURSO COMMODORE MAGAZINE
C/JEREZ, 3, MADRID-16



commodore Magazine
C/Jerez, 3 - Madrid-16

Así es la serie 700

Mucho está dando que hablar la nueva **serie 700.** Por ahora sólo han salido de fábrica los modelos **710** y **715.** El diseño de su carcasa difiere grandemente de aquellos legendarios **PET,** de sólido aspecto y múltiples aristas. Sin embargo, conservan su carácter de familia.

Se dice también que los modelos **720, 730, 740** y **750** están en fase de desarrollo. Los rumores afirman que **Commodore Business Machines** podría haber abandonado tales proyectos, pensando que por el momento es suficiente con los ya existentes. Sus 128 *Kbytes* de memoria, ampliables hasta 896 *Kbytes,* pueden dar mucho de sí. Además, los **710** y **715** soportan a los actuales estandares del mercado: **CP/M, CP/M-86** y **MS-DOS,** aunque opcionalmente.

Vistos en persona, el encanto, que pretende sugerir el aspecto que ofrecen en fotografía, se desvanece hasta cierto punto. La razón bien podría ser que el tamaño resulta algo mayor de lo que pudiera pensarse. Es similar al del **PC** de **IBM.**

A continuación vamos a describir las características comunes de la nueva serie, pues esencialmente son las mismas, salvo detalles.

Commodore presenta 2 configuraciones principales. La bautizada como "línea alta", compuesta por CPU, teclado separable y pantalla, y la "línea baja", más resumida, formada por la unidad CPU/teclado.

En principio, nos encontramos con 2 configuraciones iniciales: 128 *Kbytes* **(710 B)** y 256 *Kbytes* **(715 B).** A partir de aquí ambas son ampliables hasta un máximo de 896 *Kbytes.*

La multinacional americana no se resigna a dejar de utilizar la familia **6500** en sus ordenadores. En el caso presente ha elegido el **6509,** un versátil microprocesador. Su mayor peculiaridad consiste en la enorme flexibilidad de que dispone para direccionar memoria. El bus de direcciones emplea 16 *bits,* con lo que puede controlar hasta 64 *Kbytes.* Además, otros 4 *bits* le permiten acceder hasta a 16 bancos de memoria, de otros 64 K cada uno.

Lo que asegura a los **700** su compatibilidad con los estandares del mercado es la opción del segundo procesador. En primer lugar podría elegirse el popular **Z-80,** cargar el sistema operativo **CP/M-80** (para 8 *bits)* y correr todo el software disponible para él. La segunda posibilidad consiste en el microprocesador **8088,** el mismo utilizado por el **PC** de **IBM,** el **Rainbow** de **Digital Equipment** y muchos más, que aceptará los sistemas operativos **CP/M-86** (para 16 bits) y el **MS-DOS** y, consiguientemente, tener al alcance todo el software desarrollado para ambos.

El teclado es más completo que el utilizado por la serie **8000,** un total de 94 teclas lo asegura, ofreciendo mucho juego. La parte frontal de varias de ellas también llevan inscritos caracteres gráficos, para mayor aprovechamiento.

Un cable helicoidal lo conecta a la carcasa principal, asegurando la independencia de la unidad central.

El cuerpo principal responde a la clásica configuración QWERTY. Destaca la existencia de la tecla con la leyenda del número griego "pi".

El teclado numérico independiente está conformado por un conjunto de 19 teclas, donde aparece también la correspondiente al doble cero. La tecla CE, proporciona un rápido *"clear entry".*

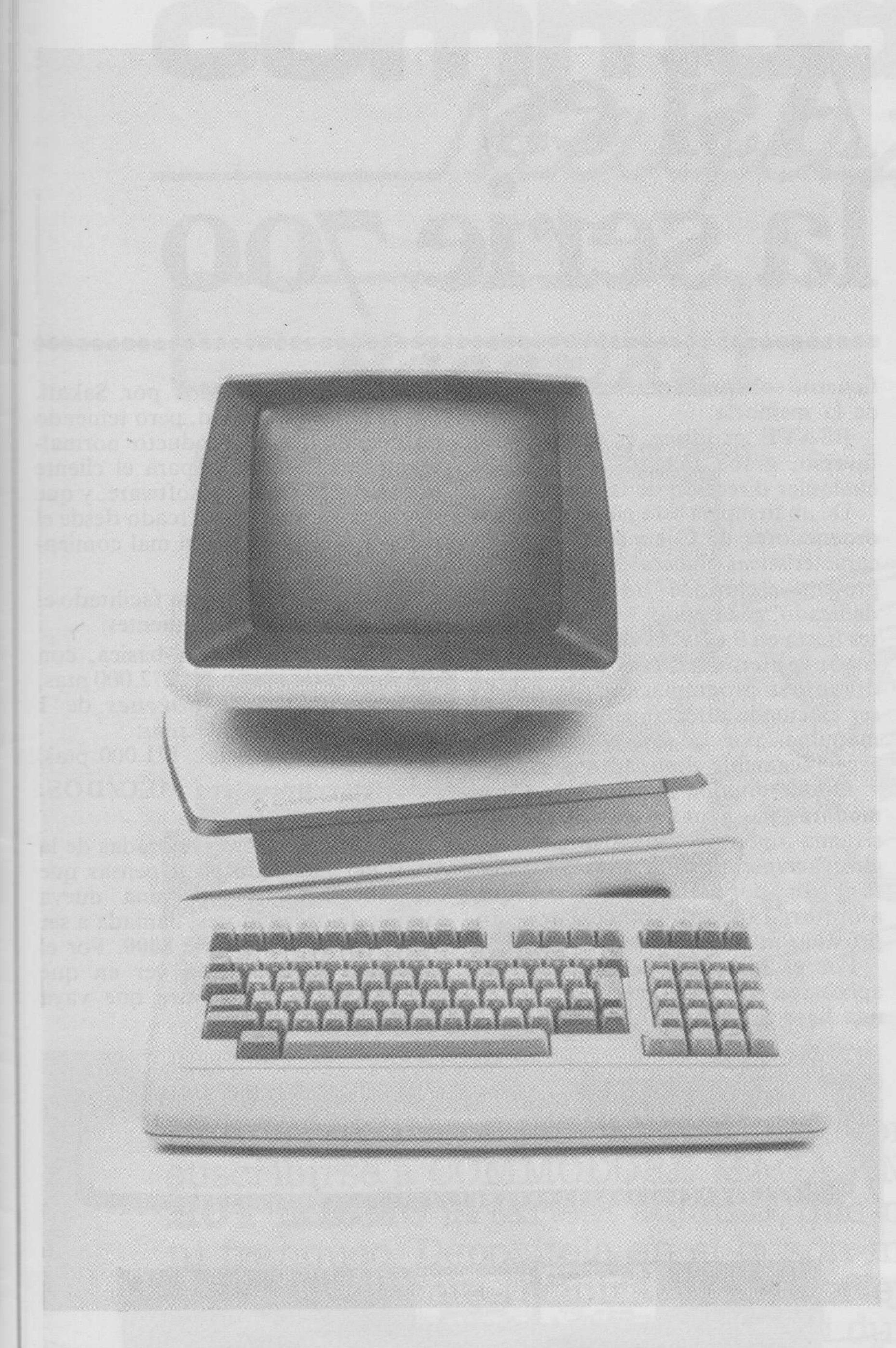
Agrupando las teclas por el fin que cumplen podemos destacar una fila de 10 teclas de función, programables por el usuario, que aportan en realidad hasta 20 funciones distintas con ayuda de la tecla SHIFT.

Otra hilera de 4 teclas permite el control directo sobre el cursor en la pantalla. De todas formas, su ubicación no parece la más cómoda para trabajar.

Para la edición en la pantalla hay 2 teclas, que aportan 4 posibilidades: INSert/DELete y CLear/HOME, constantes en las máquinas de **Commodore.** Pero también aparecen otras teclas disponibles en otros modelos: RUN/STOP (para control del *cassette),* CTRL (control), TABulación, ESCape y la tecla **Commodore.** CTRL proporciona el acceso a determinados caracteres gráficos no posibles con el uso de SHIFT. También se pueden combinar ambas.

La tecla **Commodore** detiene los listados que activan el *scroll* de la pantalla y los vuelve a liberar.

TAB tiene la clásica misión tabuladora, estableciéndose las marcas por



medio de la combinación SHIFT-TAB.

La tecla ESC no tiene desperdicio en este modelo, presionada inmediatamente antes de cualquiera de las letras alfabéticas, proporciona un nutrido conjunto de posibilidades: Borrar una línea desde la posición del cursor hasta el final, llegando a efectuar *scrolles,* y un largo etcétera.

Las teclas de mayor utilización han sido dispuestas en lugares más convenientes que los que pudieran ocupar en modelos precedentes. Estamos refiriéndonos a RETURN, ENTER (están ambas), NORMal/GRAPHico, SHIFT, SHIFT-LOCK y OFF/RVS.

El monitor utilizado se encuentra en una carcasa separada de la unidad principal. Esto permite que se le pueda hacer girar y cabecear con respecto a su base, hallando la postura más cómoda de trabajo. La pantalla tiene una diagonal de 12 pulgadas y ha recibido un tratamiento antireflectante en la superficie. El fósforo utilizado proporciona la visualización en color verde con fondo negro.

El formato adoptado también es clásico, el cómodo 25 líneas de hasta 80 columnas cada una. Los caracteres se conforman sobre una matriz de 9 por 14 puntos.

La parte posterior del equipo alberga diferentes posibilidades. En la esquina superior izquierda observamos un colorado botón, destinado a la inicialización en caliente (RESET) del sistema. Exactamente debajo de él queda situado el *interface* serie, que corresponde al estándar RS-232C, siendo además accesible a partir del intérprete del BASIC. A su lado aparece un conector DIN pentapolar para salida de video. El *slot* para conexión de la unidad de *cassette* es el siguiente, que no es un zócalo propiamente dicho, sino la misma placa de circuito impreso.

En el centro tenemos el clásico conector destinado a cartuchos.

Este *port* permite la conexión de módulos con una capacidad máxima de 24 *Kbytes,* sean de RAM, ROM o EPROM (ROM reprogramable).

Continuando el recorrido, tenemos un *jack* que actúa como salida de audio. El último *slot* que aparece es del mismo tipo que el utilizado para el *cassette,* una lengua con pistas de cobre, perteneciente a la tarjeta principal de circuito impreso. Corresponde a un *port* IEEE-488, de tipo paralelo ampliamente utilizado por **Commodore.** A través de él se pueden conectar buen número de diferentes dispositivos al ordenador. Un inconveniente podría ser que únicamente existe un *port* de este tipo, necesitándose conectar varias cosas. La solución son los conectores apilables entre sí.

Para finalizar con la descripción de

la parte posterior, solamente quedan el zócalo para el cable de alimentación desde la red eléctrica y el conmutador de puesta en marcha.

El anteriormente aludido *interface* IEE488 permite, entre otras cosas, la conexión de una gama de unidades de disco, la utilización de la impresora, etc. También se habla de la posibilidad de comunicar varios ordenadores entre sí, mediante una red.

El único lenguaje de programación disponible para los **700,** por ahora, es el BASIC interpretado y extendido. Con extendido se entiende que dispone de comandos no clásicos en el BASIC más o menos estándar. En este caso es un nutrido conjunto.

Otros lenguajes de alto nivel son posibles, cuando se trabaja con los sistemas operativos estándar y el segundo procesador, por haber sido desarrollados para trabajar conjuntamente. De todas maneras, **Commodore** anunció otras opciones, tales como el **Pascal, Forth, Prolog** y **Logo,** aunque por el momento no están disponibles. Retornando al BASIC, destaquemos que el intérprete va alojado en 28 de los 40 *Kbytes* totales de la ROM.

BANK facilita la escritura de programas que necesiten más de los 64 *Kbytes* de un banco de memoria, asignando el banco de trabajo en cuestión.

DELETE borra un número de líneas introducidas previamente en la memoria.

DISPOSE aborta un bucle FOR... NEXT o GOSUB... RETURN si se ha producido un error.

ERR$ (n) devuelve el mensaje de error correspondiente al número indicado.

Una tecla de función se asocia con un texto mediante KEY n, "texto", y PUDEF permite redefinir ciertos símbolos del formato asignado por PRINT USING.

TRAP facilita la bifurcación de una línea indicada, si se produce un error durante la ejecución del programa, devolviéndose el control por medio de RESUME, una vez corregido.

También existen comandos destinados al acceso a disco desde BASIC. Entre ellos encontramos BLOAD, que se encarga de la carga binaria de ficheros sobre cualquier zona elegida de la memoria.

Así es la serie 700

BSAVE produce el mecanismo inverso, graba ficheros a partir de cualquier dirección de la memoria.

De un tiempo a esta parte, todos los ordenadores de **Commodore** aportan características musicales. En el caso presente, el chip *6581* trabaja, de modo dedicado, generando 3 voces diferentes hasta en 9 octavas de amplitud. El inconveniente podría presentarse durante su programación, que deberá ser efectuada directamente en código máquina, por no existir comandos específicamente destinados a tal fin.

El distribuidor nacional de **Commodore** en España distribuye un sistema operativo, desarrollado exclusivamente para la serie **700,** que responde por **MEC/DOS,** del que ampliaremos información en un próximo artículo.

Por el momento, el software de aplicación que trabaja con el **710** es una Base de Datos y un Tratamiento de Textos, distribuidos por Sakati. No es mucha cantidad, pero teniendo en cuenta que el producto normalmente se personaliza para el cliente por parte de casas de software, y que sólo está en nuestro mercado desde el pasado SIMO, no es un mal comienzo.

Los precios que nos ha facilitado el distribuidor son los siguientes:

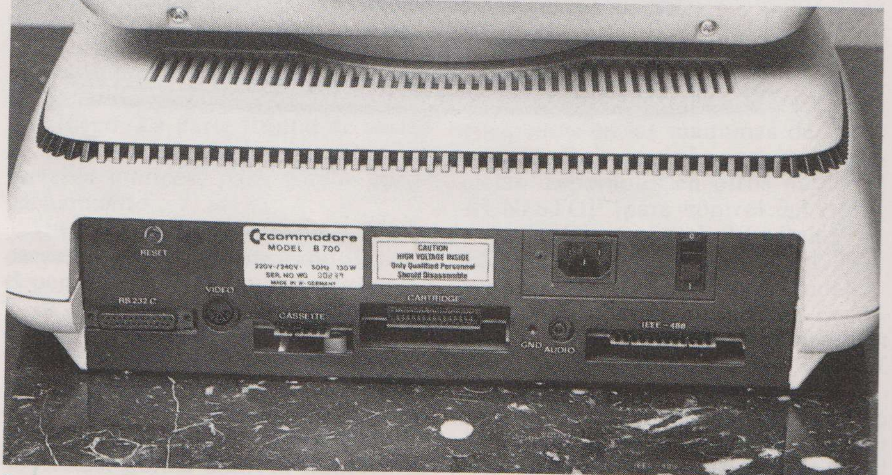
710 en configuración básica, con 128 *Kbytes* de memoria: 272.000 ptas.

Doble unidad de *diskettes* de 1 *Mbyte* (8250): 300.000 ptas.

Impresora matricial: 171.000 ptas.

Sistema operativo **MEC/DOS:** 25.000 ptas.

Las características mejoradas de la serie **700** nos inducen a pensar que nos encontramos ante una nueva familia de ordenadores, llamada a ser digno relevo de la serie **8000.** Por el momento esperamos a ver en qué línea se sitúa el *software* que vaya apareciendo.



Parte posterior del 710.

VicTree, el árbol de los comandos

Cuando pretendemos obtener el máximo rendimiento de un ordenador, nunca está de más disponer de nuevos comandos especializados que nos faciliten la tarea. Esto es más o menos lo que viene a ofrecernos el **VicTree.** Un diminuto cartucho, enchufable al **VIC-20** ó **Commodore 64**, nos proporciona un nuevo conjunto de 42 comandos básicos que, sumados a los propios, multiplican la potencia de nuestra programación y la hacen más eficiente.

El cartucho simplemente se conecta al *port* de expansión del ordenador, lo demás será únicamente leerse el manual.

El fabricante, la empresa californiana **Skyles Electric Works,** ha preferido definirlo como una potente caja de herramientas que soporta todos los comandos de BASIC 4.0 (compatible con la unidad de disco), comandos para facilitar el control de la impresora, comandos para la edición y comandos para la depuración de programas. Además, se promete hacer posible el acceso a una red local conocida por Cee-Net.

El manual es cauto. Comienza advirtiendo la suposición de que el usuario está "razonablemente familiarizado" con uno de los 2 ordenadores de **Commodore.** Añadimos que es obvio que un principiante no esté preocupado por localizar herramientas de ayuda a la programación (anoten la palabra *software tools*, se pondrá de moda).

El objeto del presente artículo es la exposición de las ventajas que podría conllevar el módulo. Sin embargo no vamos a ser excesivamente prolijos en su descripción. Destacaremos solamente aquello que nos parezca de especial interés, pretendiendo que sea el propio lector quien dictamine sobre su utilidad en particular.

La conexión del módulo es simple. La única precaución a tener en cuenta es la consabida de asegurarse que el ordenador no está recibiendo alimentación. Una vez enchufado, se actúa sobre el conmutador ON-OFF, encendiéndose el *LED* (diodo emisor de luz), como de costumbre. Según se disponga de un **VIC-20** o un **CBM 64**, la variación es mínima. En el primer caso, como es normal, la pantalla mostrará el siguiente mensaje: CBM BASIC V2, 2951 BYTES FREE, SUPER EXPANDER, READY. En este momento ya se puede teclear: SYS 45056: CLR. El ordenador responderá con: VIC-TREE, (C) 1982 ROB CHANG, READY.

Lo cierto es que la cantidad de memoria disponible será diferente, según el módulo utilizado.

Para el **64,** después de aparecer el mensaje: COMMODORE 64, BASIC V2, 64 K RAM SYSTEM 30719 BASIC BYTES FREE, READY, se introducirá: SYS 32768. La respuesta es la misma que en el caso del VIC.

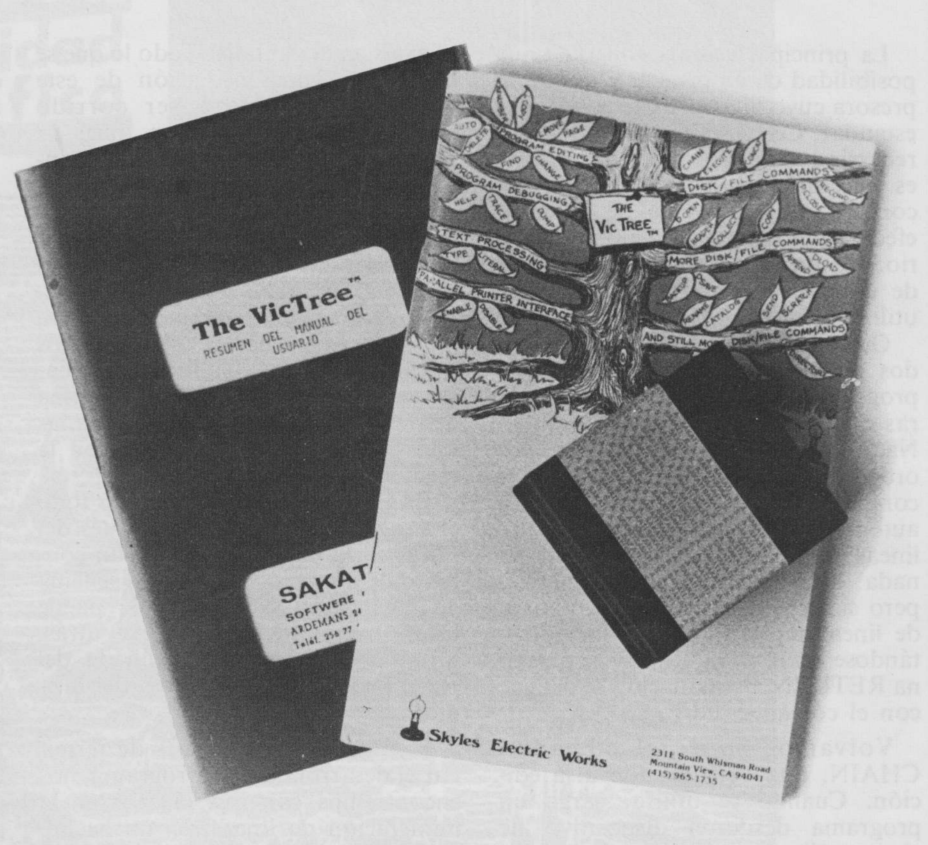
Pudiera ser que el usuario piense dejar de utilizar el VicTree sin desenchufar el cartucho. Simplemente con KILL (muerte) y RETURN, desaparece, sin dejar huella, en el intérprete del BASIC y el sistema operativo.

No hemos aclarado que no sirve el mismo módulo para ambos ordenadores. El diseñado para el **VIC-20** dispone internamente de 8 *Kbytes* de ROM y 4 *Kbytes* de memoria RAM,

de los cuales 1 *Kbyte* está reservado a los comandos del **VicTree.** El resto de los 4 K se reserva a los *buffers* para comunicación con la red Cee-Net, que permite que puedan ser conectados hasta 64 **VIC-20** o **CBM 64,** entre sí, compartiendo impresoras y unidades de disco.

La ROM reside en las direcciones 45056 (B000) a 49151 (BFFF) y 28672 (7000) a 32767 (7BBB) y la RAM entre el 24576 (6000) y 28671 (6FFF), en el **VIC-20.**

En el **64,** la ROM es de 8 *Kbytes,* situados entre las direcciones de memoria 32768 (8000) y 40959 (9FFF). La RAM también es de 4 *Kbytes,* situándose entre 49152 (C000) y 53247 (CFFF). Los programas que necesiten utilizar las mismas direcciones de memoria que emplea la RAM de este cartucho, obligarán a desactivar el **Vic-Tree** para correr sin problema.



Todos los comandos de gestión del diskette utilizados por el **Vic-Tree** tienen una sintaxis compatible con el BASIC 4.0 de **Commodore.**

El conjunto de comandos destinados al control de los discos es amplio, 21 en total, la mitad de los que proporciona el **Vic-Tree.** Ellos posibilitan la utilización de cualquiera de las diferentes unidades. La función de 15 de estos comandos es la misma que sus homólogas del BASIC 4.0, el resto son nuevos.

Podemos destacar el comando EXECUTE, cuya misión consiste en aglutinar LOAD y RUN de una sola vez.

Para la copia de disco a disco, empleando 2 unidades, aparecen 2 comandos. BACKUP copia el *diskette* completo, facilitando la generación de copias de seguridad. COPY copia únicamente ficheros, transfiriéndolos de un *diskette* a otro. La diferencia entre ambos comandos estriba en que COPY no destruye ficheros previamente existentes en el *diskette.* La única limitación que aparece es la no identidad de nombre entre ficheros.

Para visualizar en pantalla el nombre de los ficheros contenidos en un *diskette* existen dos comandos: DIRECTORY y CATALOG, son el mismo, pero el primero destaca por ser el mismo nombre empleado por el BASIC 4.0.

Interesante es la posibilidad ofrecida por CHAIN#, un comando que ejecuta la carga (LOAD) bajo el control de un programa. La aplicación más inmediata se encuentra en los programas, que por su excesiva longitud, no pueden ser cargados de una sola vez en la memoria central del ordenador. Entre los comandos destinados a la edición aparece otra vez CHAIN, el cual pone a cero el área de almacenamiento de las variables. Con CHAIN# no aparece tal problema, pudiendo compartirse el mismo juego de variables. El acceso a los registros queda previsto por el comando RECORD, que hace que la unidad de disco prepare un registro para ser escrito o leído.

El **VicTree** permite al usuario definir sus propios comandos para utilizar desde el BASIC. No es sencillo, se requiere un extenso conocimiento del lenguaje máquina y la arquitectura interna del intérprete BASIC del ordenador en particular. Pero es algo a tener en cuenta. Estos nuevos comandos, una vez definidos, hay que conservarlos en disco, para su posterior utilización. En 6 páginas del manual se encuentra una buena y útil descripción de los ficheros en disco. Comienza diciendo que "Un fichero de datos es un conjunto de informaciones organizadas en forma de líneas o caracteres individuales". Posteriormente trata 2 tipos diferentes de los mismos: los secuenciales y los relativos. Los ejemplos que acompañan son lo suficientemente explícitos para lograr una idea precisa sobre su utilidad. Recalquemos que los ficheros relativos son realmente cómodos en la práctica, sobre todo cuando se trata con grandes cantidades de datos.

Los comandos aportados por el **VicTree** para control de la impresora son únicamente 2: ENABLE y DISABLE. Con el primero se advierte a la impresora para que se disponga a ser utilizada. El otro hace lo contrario, la desactiva.

VicTree

La principal ventaja consiste en la posibilidad de emplear cualquier impresora cuyo interface corresponda al estandar Centronics, sin tener que recurrir a ningún otro hardware. Sólo es necesario un trozo de cable. El control y transferencia de datos se efectúa por el *port* paralelo del usuario. No queda mucho más por decir de esta posibilidad. Simplemente se utiliza o no.

Otro nutrido conjunto de comandos es el destinado a la edición de programas. Ofrece sustanciales mejoras con respecto al BASIC estándar. Nada más comenzar su revisión, por orden alfabético, nos encontramos con AUTO, que numera de forma automática (de ahí su nombre) las líneas de programa, de manera ordenada. Se pueden definir parámetros, pero si no se hace, el primer número de línea empleado es 100, incrementándose en 10 cada vez que se presiona RETURN. Su actuación se detiene con el comando OFF.

Volvamos sobre el comando CHAIN, esta vez aplicado a la edición. Cuando se utiliza, carga un programa desde el dispositivo de cinta o disco, situándolo detrás del programa previamente existente en la memoria del ordenador. Sin embargo, hay que tener en cuenta que este comando no altera los números de línea, por tanto si no se cuida la numeración, los programas se enmarañaran entre sí.

CHANGE busca algún texto incluido en un rango dado de líneas y, cuando lo encuentra, pasa a sustituirlo por otro especificado. DELETE borra de la memoria del ordenador un rango deseado de líneas de programa.

El comando FIND tiene una utilidad parcialmente similar a CHANGE. Se limita a localizar texto o palabras empleadas por el BASIC, con sólo definirlo tras el comando. Una vez finalizada la búsqueda, el resultado será visualizado en pantalla. El comando LITERAL resulta curioso, activa un editor de tratamiento de texto, permitiendo la posibilidad de almacenar texto directamente en la memoria del ordenador, sin obtener el mensaje de error de sintaxis en la pantalla. Todo lo que se introduzca, por mediación de este comando, no podrá ser corrido (RUN) por el ordenador, pues se prevé que no se han utilizado comandos del BASIC.

Una segunda, y no menos importante, aplicación de LITERAL es su actuación de un modo idéntico al comando AUTO.

Cuando se desarrolla un programa largo, la existencia de MERGE nos facilitará las cosas. Su funcionamiento consiste en cargar un programa desde un disco y entrelazarlo con las líneas de programa existentes en la memoria.

PAGE posibilita "saltar" por todo el programa a elección. Después del salto, en la pantalla aparece un conjunto de líneas, como si de un sistema de paginación se tratara. Se puede listar hacia adelante, hacia atrás, saltar a una línea determinada del programa o volver al inicio del mismo.

Es normal que, después de terminar el desarrollo de un programa, nos encontremos con que el caos en la numeración de líneas sea un hecho. En tal caso puede ser deseable renumerarlas en base a un determinado incremento. Esto es precisamente lo que hace el comando RENUMBER. Puede elegirse el clásico 100 para la primera línea y el incremento de 10 sucesivamente, o dar un número de comienzo y otra cantidad para el incremento. La renumeración tampoco tiene porque afectar a todo el programa, puede reducirse a un rango elegido.

El último comando de edición resulta bastante interesante. TYPE sería equivalente a RUN, pero con los textos. Se limita a "listar" el documento previamente introducido con el tratamiento de texto, sin aparecer números de línea. Se pueden especificar por el usuario el margen izquierdo y la extensión que se desea visualizar, pudiendo ser enviado directamente a la impresora.

Los comandos para la depuración de programas son 3. Su misión se centra en localizar y diagnosticar errores de programación.

DUMP visualiza en pantalla las variables, que no sean cadenas, que en un momento dado tienen asignado un valor, pudiendo alterarse a voluntad.

Por último, TRACE es un comando que existe en otros modelos de ordenador, con un cometido similar. Permite seguir el orden de ejecución de las líneas del programa, observándose el programa paso a paso. La línea que va a ser ejecutada aparece en la parte superior de la pantalla. Mediante OFF se desactiva TRACE.

La red local (LAN) **Cee-Net** utiliza simplemente un cable bifilar para interconectar hasta 64 ordenadores y sus periféricos. El protocolo utilizado para evitar conflictos en la red es del tipo *Carrier Sense Multiple Access* (algo similar a acceso múltiple por sentido de la portadora), que detecta la colisión entre mensajes y errores producidos durante la transmisión de datos. Las informaciones se colocan en la red en forma de paquetes de datos. Utilizando el comando ATTACH, el **Vic-Tree** intenta establecer comunicación mediante la carga de subrutinas de alto nivel, que proporcionan un interface del BASIC con la red. Si el resultado es positivo, se puede indicar con qué dispositivo, conectado deseamos comunicar.

El manual que acompaña al módulo es razonablemente comprensible. No se extiende en rebuscadas explicaciones. El índice también está perfectamente documentado, por lo que se convierte en gran ayuda durante el aprendizaje. No se hace necesario buscar en diferentes páginas para conocer la utilidad de un comando. Tuvimos oportunidad de leer también el manual traducido al castellano que **Sakati, S. A.,** firma que comercializa el **Vic-Tree** en nuestro país, ha elaborado acertadamente.

La conclusión a que llegamos, después de revisar el funcionamiento de **VicTree,** es que el fabricante lleva razón. Se trata de una verdadera herramienta de ayuda a la programación. Su utilización es relativamente sencilla. Si cae en manos de un profano se hacen cosas, pero si es un programador creativo quien lo usa, podrá obtener un enorme rendimiento.

El ordenador que controla

Que un ordenador puede procesar datos es cosa que a nadie sorprende. Pero si alguien afirma que un ordenador es capaz de controlar las evoluciones de un tren eléctrico, encender la cafetera a las 9 o conectar y desconectar bombillas, siguiendo un orden predeterminado, puede producir la hilaridad en mucha gente.

Sin embargo tal aseveración es cierta. El ordenador no sólo puede hacer tales cosas. Cuando se trata de dispositivos alimentados por electricidad, se puede programar en que momento reciben alimentación.

El lector se preguntará: Bueno, ¿y cómo lo hace? Aunque a primera vista parezca un misterio, el ordenador está en disposición de proporcionar una serie de señales de control que, adecuadamente tratadas, controlarán el paso de la corriente.

El dispositivo, tema de este artículo, convierte en realidad las posibilidades descritas. Su nombre comercial es **Elektrocomputer**, pudiendo ser enchufado (viene en forma de cartucho) tanto al **VIC-20** como al **CBM 64**.

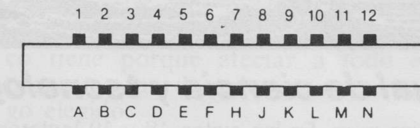
La descripción del producto será más comprensible si conocemos de antemano algunos de los principios, que aclaran como se comunican ambos microordenadores con el mundo exterior (de cara a los fines que nos interesan).

Es por todos sabido que un componente clave de los microordenadores es el microprocesador, un *chip* en el que converge finalmente el tratamiento de los datos. Como es de suponer, el **VIC-20** y el **CBM 64** poseen el suyo, que, además, es el mismo. Su nombre de guerra responde a 6502, uno de los más ampliamente utilizados, incorporándolo hasta el legendario Apple.

Para ser rigurosamente exactos, aclaremos que el 64 lleva el 6510, que no es ni más ni menos que una versión del 6502, que dispone de capacidades para el CIA en el mismo encapsulado. Las siglas responden al variopinto nombre de *Complex Interface Adapter*, algo así como adaptador para *interface* complejo. La misión encomendada a él consiste en dedicarse, de manera especializada, a la comunicación con dispositivos externos.

En el caso del **VIC-20**, el papel equivalente lo protagoniza otro circuito integrado, el 6522, que tiene por nombre VIA *(Versatil Interface Adapter)*. El **64** dispone además de otro *chip* que amplía las posibilidades del CIA, llamado 6526.

Figura 1. *Port* del usuario. La misma configuración sirve para el **VIC-20** y el **CBM 64**. Debajo la correspondencia entre terminales.



PATILLA	CORRESPONDE
2	+5 V
A	MASA
C	PB0
D	PB1
E	PB2
F	PB3
H	PB4
J	PB5
K	PB6
L	PB7
N	MASA

Comprendiendo el funcionamiento del VIA 6522, tendremos las ideas lo suficientemente claras para entender la forma en que el ordenador se comunica con los dispositivos periféricos.

El 6522 proporciona 16 registros destinados a diversas funciones de Entrada/Salida. En el caso que nos ocupa va directamente conectado a la ranura del *port* del usuario, que en ambos modelos se sitúa en la parte posterior del ordenador, justo en el margen derecho.

Por diseño, el VIA proporciona 2 *ports*, PA y PB respectivamente según las especificaciones del fabricante. Dependiendo de qué ordenadores se utiliza uno u otro. En el **VIC** y el **64** es el B.

El *port* consiste fundamentalmente en 8 líneas paralelas, conectadas directamente al *port* de usuario del ordenador (en la figura son PB0 a



PB7). Lo más interesante de ellas es que son programables de forma individual, de tal manera que cada bit puede ser de entrada o de salida. Aclaremos. Cada una de las patillas PBX puede ser definida para que tome la señal desde el periférico y la introduzca en el ordenador (entrada), o para enviar el bit desde el ordenador al exterior.

Otra pregunta surge en torno a cómo y quién decide qué bit del *port* se comporta de que manera. Se contesta diciendo que se programa por software. Existe en el VIA (o CIA) un registro bautizado DDR, según su nombre en inglés, y registro de dirección de datos en traducción libre. En él se guarda la información sobre qué dirección han de tomar los datos en cada momento. Cada línea de datos del *port* está representada en el DDR. Si, en él, un bit es 1, la línea correspondiente en el *port* responderá como salida. En cambio si es 0 actuará como entrada. Dependiendo de la información que pongamos en el DDR, podemos configurar el *port* como más nos venga en gana.

Si, como en el caso presente, lo que se pretende es controlar un dispositivo desde el ordenador, habría que poner a 1 los 8 bits del DDR.

Ambos registros son controlables a partir de una dirección de la memoria RAM del ordenador. La única diferencia entre el **VIC** y el **64** estriba en la dirección. Así, el mecanismo consiste en poner esta información en las distintas direcciones de cada máquina, en forma de bytes. La utilización de **POKES** desde el **BASIC** proporcionan total ductilidad en el control.

Para el **VIC-20,** encontramos que el DDR está situado en la dirección 37138 (decimal) y la 56579 para el **CBM 64.** El registro de Entrada/Salida queda situado en la dirección 37136 para el **VIC** y 56577 para el **64.**

El sistema **Elektrocomputer** no necesita recibir más información que la que expresa si cada uno de los 8 bits es activo o inactivo (1 ó 0), para de esta forma controlar que el dispositivo exterior, asociado a ese bit reciba o no alimentación. Para ello, no utiliza los 2 registros, sino que toma directamente la información depositada en el DDR, simplificando el programa de control que necesita el usuario. El registro de E/S no se hace necesario en tal caso.

Una vez tomado el byte de información desde el ordenador, el **Elektrocomputer** procesa las señales de cara a gobernar 8 relés. Aclaremos nuevamente que el resultado final es que cada relé depende de un bit.

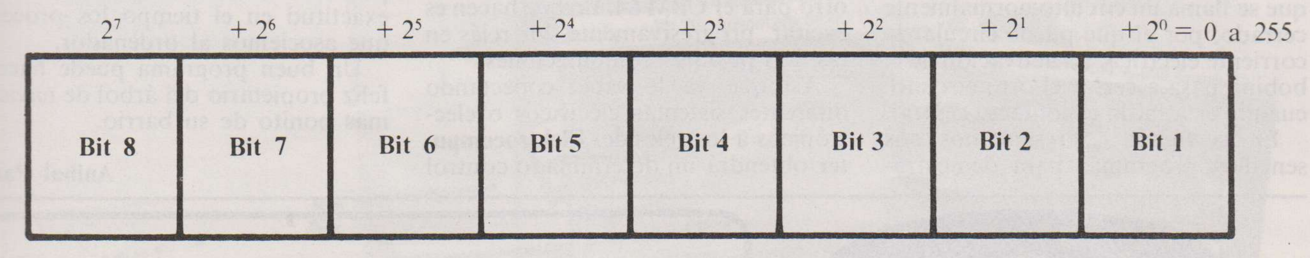


Figura 2. Peso de cada *bit* con valor 1 en el total del *byte.*

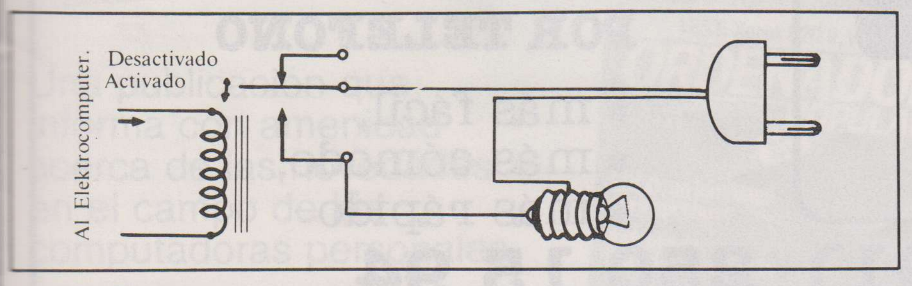


Figura 3. Diagrama esquemático del funcionamiento de un relé.

La forma en que el ordenador accede al **Elektrocomputer** va implícita con el POKE 37138,X ó POKE 56579,X. Simplemente con esto, el byte X es transferido y almacenado en él. Otro POKE posterior conteniendo un nuevo byte X proporciona las nuevas órdenes.

Sabemos que un byte está compuesto por 8 bits, cuyo valor es más significativo cuanto más a la izquierda se encuentre, tal como ocurre con los números decimales. Si por ejemplo queremos activar solamente el primer relé, el byte tomará la forma 00000001. Si, por el contrario, queremos activarlos todos, el byte será 11111111.

Realmente, los unos y ceros del byte tienen un equivalente en numeración decimal, 00000001 corresponde a 1 y 11111111 vale 255. Luego existen 256 posibles combinaciones.

La forma más sencilla para decidir que relés van a ser activados y cuáles no, es elaborar la configuración de unos y ceros, de acuerdo a nuestro interés, convertirlo en decimal y ponerlo en el lugar ocupado por la X en

los POKES arriba mencionados. El valor de cada bit viene expresado por una potencia de 2, como ilustra la figura 2. Si queremos activar los circuitos 2 y 6, el valor de la X ha de ser 2 + 32 = 34. Sumando el valor equivalente de cada 1 obtendremos el número que hay que depositar en el DDR.

Revisar el concepto de relé puede ser de interés para algunos lectores. Este útil dispositivo electromecánico se basa en un principio simple. Una corriente eléctrica circula por la bobina de un electroimán, atrayendo este último a un contacto eléctrico (ver figura 3). En la posición de reposo cierra un circuito. Posee lo que se llama un circuito normalmente cerrado, por el que puede circular la corriente eléctrica. La activación de la bobina pasa a cerrar el otro circuito, cuando es atraído el contacto central.

En la figura 4 presentamos dos sencillos programas para demostración del funcionamiento del **Elektrocomputer**, uno para el **VIC-20** y el otro para el **CBM 64.** Lo que hacen es excitar, progresivamente, los relés en las 256 posibles combinaciones.




```
10 FORX=1 TO 255
20 POKE37138,X
30 FORA=0 TO 100:NEXTA
40 NEXTX
```

VIC-20

```
110 FORX=1 TO 100
120 POKE56579,X
130 FOR A=1 TO 100:NEXTA
140 NEXTX
```

CBM 64

Figura 4. Programas de test utilizados para comprobar el Elektrocomputer. Para el **VIC-20** y el **CBM 64** respectivamente.

Así que ya lo sabe, conectando diferentes sistemas eléctricos o electrónicos a los relés del **Elektrocomputer** obtendrá un determinado control sobre ellos. Aprovechando el TI$ podrán secuenciarse con una cierta exactitud en el tiempo los procesos que asociemos al ordenador.

Un buen programa puede hacerle feliz propietario del árbol de navidad más bonito de su barrio.

Aníbal Pardo

Iniciamos una sección permanente, destinada a comentar y analizar críticamente los programas existentes en nuestro mercado, para correr en los ordenadores de ***Commodore****. En algún caso, podremos incluir programas que no estén aún presentes, bien por su interés para nuestros lectores, o porque estén próximos a aparecer.*

La revisión ha sido confiada a experimentados profesionales del software (para estas y otras máquinas), por lo que sus criterios, aunque puedan parecer subjetivos, son lo suficientemente ponderados que un profesional está en condiciones de tener. De todas maneras, aquí van las principales pautas observadas:

Cuando tratemos con juegos, en primer lugar se evaluan las cualidades que el programa posee para retener la atención del jugador, es decir, su capacidad para crear adicción. En segundo lugar, se analiza la presentación formal, tanto en aspecto externo como el contenido de la pantalla. Tercero, se califica la calidad de la representación gráfica conseguida. Por último, se considera la cantidad de acción y movimiento generado durante el juego.

Cuando analizamos programas de aplicación, sea el caso de una contabilidad, por ejemplo, se partirá evaluando su utilidad real. En segundo lugar, la presentación formal, igual que en los juegos. En tercero pasa a puntuar la claridad y facilidad de manejo. Para finalizar, la rapidez de ejecución es el factor que cuenta.

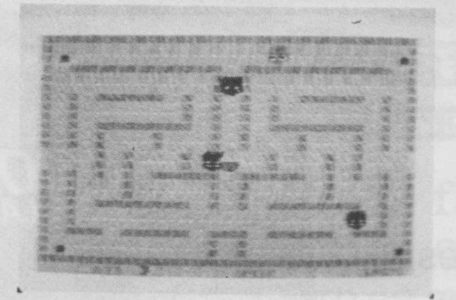
PROGRAMA: SPRITE MAN
TIPO: JUEGO
DISTRIBUIDOR: INDESCOMP
FORMATO: CINTA DE CASSETTE COMMODORE 64

CALIFICACION ■ ■ ■ □ □

Este programa es una versión del popular comecocos, realizada para el **Commodore 64**, por lo que no es necesario dar más explicaciones sobre el fin básico del juego: comerse los puntos y evitar a los fantasmas.

Las escuetas instrucciones que vienen en la caja son suficientes para cargar el programa y ponerlo en marcha. Asimismo, se incluyen las instrucciones de juego, pudiendo hacerlo con un *joystick* conectado al *PORT 2* o mediante el teclado, siendo en este caso las teclas de movimiento: 3 para arriba, X para abajo, para la izquierda y para la derecha. A mi modo de ver esta es una disposición bastante incómoda y sería mejor que las teclas de direcciones opuestas estuviesen más juntas, como por ejemplo, A para arriba y Z para abajo. De todas formas sobre gustos no hay nada escrito.

Otro fallo que le veo (si se puede llamar así) es que hay que pulsar la tecla de cambio de dirección justo en el momento en el que lo vayamos a realizar. Ni un poco antes ni un poco después, dando un margen demasiado pequeño —que nos impide concentrarnos en el juego propiamente dicho— para estar atentos al momento del cambio. Esta operación me pareció más crítica cuando se usa el teclado (aunque puede ser una impresión subjetiva). El juego en sí es bastante entretenido; la forma del



laberinto difiere de la que estamos acostumbrados a ver en los bares. Las famosas frutas de bonos también están incluidas en este juego, pero con la particularidad de que aparecen al principio de la pantalla correspondiente y están encerradas, pudiendo acceder a ellas más adelante, cuando dos de las paredes que las encierran desaparecen. Hay que tener cuidado con los fantasmas, porque la única diferencia entre el estado en que uno se los puede comer y cuando ellos te comen está en los ojos, y si no se va con cuidado puede cometerse una equivocación fatal.

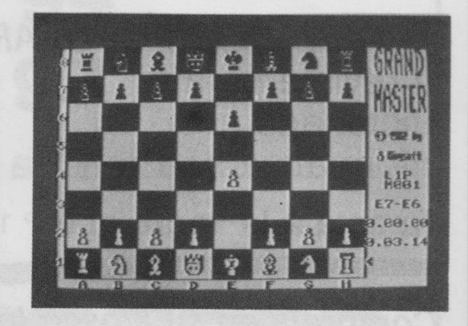
El programa tiene una presentación atractiva, aunque no genere sonidos que acompañen al juego y da la diversión que se espera de él.

PROGRAMA: GRAND MASTER
TIPO: JUEGO
DISTRIBUIDOR: ABC ANALOG
FORMATO: CINTA DE CASSETTE COMMODORE 64

CALIFICACION ■ ■ ■ ◪ □

Dentro de los juegos de inteligencia el rey es, sin ninguna duda, el ajedrez. Es el que más obliga a pensar para jugar correctamente. Por esta misma razón, su implantación en los ordenadores ha sido siempre difícil y no siempre con buenos resultados. Pero en este programa se ha logrado meter un buen jugador dentro de una máquina de unas dimensiones relativamente reducidas.

El programa viene con unas instrucciones desusadamente buenas, llegando incluso a introducir al final una corta explicación de las bases del juego.



Tiene diez niveles de juego, que van desde el ultrarápido (con cinco segundos de pensamiento) hasta el lento (dos horas por jugada), siendo el nivel cero especial para análisis de

jugadas por correo. Además de estas características, dispone de otras cualidades como son: Juego automático contra él mismo, para poder aprender la estrategia del juego, cambio de los colores (no muy útil pero sí bonito), elección del interlazado (característica que algunos televisores necesitan). Aunque no parezca útil muy a menudo (la persona que lo necesite lo agradecerá), retroceso de una jugada errónea, cambio de las fichas al principio para que empiece a jugar el ordenador (no todos los programas lo tienen), tutor de ajedrez, que aconseja sobre la mejor jugada), etcétera.

La máquina juega realmente bien y sin fallos, ofreciendo partidas agradables a toda persona (novel o experimentada) que lo juegue.

Tiene una presentación muy vistosa.

PROGRAMA: WHO DARES WINS
TIPO: JUEGO
DISTRIBUIDOR: ABC ANALOG
FORMATO: CINTA DE CASSETTE COMMODORE 64

CALIFICACION ■ ■ ◩ □ □

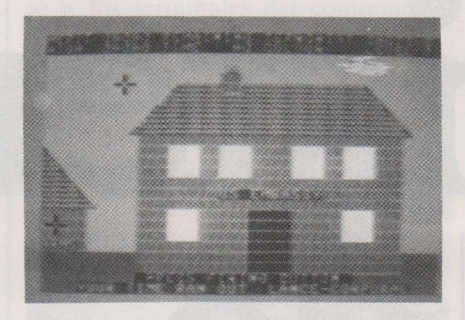
El terrorismo también ha llegado a los ordenadores que, en este juego, nos convierten en miembros de las SAS (servicios especiales Británicos), encargados de rescatar al personal de la embajada estadounidense que ha sido secuestrado.

La carga del juego es sencilla y se debe señalar que todas las instrucciones y los mensajes del juego van en inglés, aunque el distribuidor nos ha informado que van a sacar una traducción, que se entregará con el juego tan pronto como esté disponible.

El juego en sí dibuja un edificio, donde se encuentran los secuestradores y los rehenes. El jugador dispone de una mirilla telescópica, controlada con el *joystick* (no hay versión para el teclado) con la que se debe apuntar y disparar a los terroristas cuando aparezcan por una ventana. El modo de diferenciar a los terroristas de los rehenes es guiándose por el color de las camisetas. En la parte superior de la pantalla hay un rectángulo, que indica el color de la camisa de los asaltantes, cambiando este color esporádicamente.

En la misma parte superior se indican los puntos conseguidos y el tiempo (tenemos un tiempo limitado para matar a los terroristas).

El juego tiene nueve niveles, siendo la principal variación de nivel a nivel la velocidad con que se mueven las personas, y si el nivel 1 (el más sencillo) es difícil, el nivel 9 es para



auténticos francotiradores. Cuando se nos acaban las balas (indicado también en la parte superior) tenemos que colocar la mirilla exactamente sobre una cruz que hay a la izquierda del edificio, recargándose el fusil inmediatamente.

El juego está bien desarrollado y es una prueba de reflejos, aunque le falta algo de excitación, y a la larga termina siendo monótono.

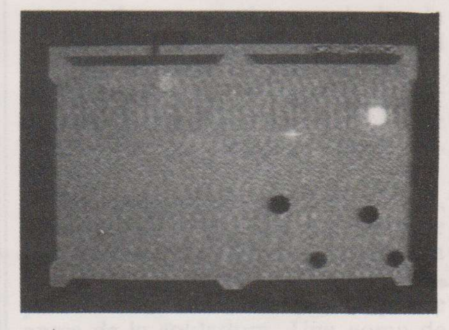
PROGRAMA: HUSTLER
TIPO: JUEGO
DISTRIBUIDOR: INDESCOMP
FORMATO: CINTA DE CASSETTE COMMODORE 64

CALIFICACION ■ ■ ■ □ □

Este juego es una bonita simulación de un billar americano, en versión reducida (son seis bolas además de la blanca), y nos ofrece un juego distinto de los típicos marcianitos. No hay que andar con prisas porque alguien nos vaya a comer.

Las instrucciones de carga que vienen en la caja son claras y no dan ningún problema. Para jugar se puede utilizar un *joystick* o el teclado. Aunque en este último caso se indica qué teclas son las que hay que usar, pero no para qué vale cada una. En ambos casos el usuario mueve una cruz por la mesa. Esta cruz indica la posición hacia donde va a ir la bola blanca, partiendo de su posición actual. Una vez que hemos indicado el punto deseado, pulsamos el botón del *joystick* (o la barra del espacio) y la bola avanza produciendo los rebotes.

Aunque no se indique en las instrucciones, hay que explicar que en la parte inferior derecha hay una barra que se desplaza, indicando la fuerza con que sale lanzada la bola. Para lanzar con una fuerza deseada sólo hay que esperar a que la barra indi-



que la fuerza que deseamos y apretar en ese momento.

Pueden jugar una o dos personas y posee las tres variantes típicas: cualquier bola en cualquier agujero, introducir las bolas por orden en cualquier agujero y cada bola en su agujero. En caso necesario el programa numera las bolas, para saber cuál es cada una. También guarda una lista de los mejores récords (en golpes) conseguidos desde que se encendió la máquina.

La presentación está conseguida en base a bonitos gráficos y efectos muy bien logrados.

El misterio de soldar

Intentaremos que, habitualmente, aparezca en estas páginas un montaje electrónico destinado a los ordenadores de Commodore. El objetivo de esta sección es permitir a quienes deseen "cacharrear" con su micro, que puedan dar rienda suelta a su afición.

Algunos lectores, interesados en el tema pero no suficientemente formados, nos han hecho notar que estábamos apuntando demasiado alto. ¿Por qué no habrían de aprovechar esta sección los lectores que no presumen de manitas? Por tanto, hemos decidido publicar periódicamente, entre montaje y montaje, artículos de iniciación.

A no ser que uno utilice esas caras tarjetas que facilitan los montajes provisionales, se impone aprender a soldar con propiedad. Las *meigas,* que tienen la virtud de fastidiarnos los montajes, desaparecerán con el simple encanto de la soldadura bien hecha.

No se confíe usted creyendo que soldar es un juego de niños, que no consiste más que en aplicar los grados

de calor adecuados al estaño y ya está. Fundir el estaño no es más que una parte del proceso total.

Una buena forma de aprender a soldar es observar a alguien que haya conseguido una cierta maestría en el tema. Después se hará lo propio bajo la atenta mirada del maestro.

Como no todos tenemos ese amigo dispuesto a enseñarnos, veremos unas cuantas líneas maestras prácticas para hacer soldaduras fiables, sin por ello freír los componentes, ni los dedos.

Una norma, aunque no tiene por qué cumplirse siempre, es que la soldadura que tiene un aspecto bonito, suele ser correcta.

Es conveniente, como primer paso, hacernos con una serie de útiles y herramientas, de las cuales lo más importante son el soldador y el estaño.

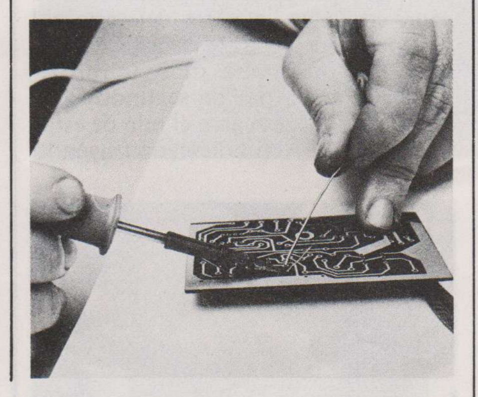
El tipo de circuitería que vamos a soldar sobre placa de circuito impreso, o las tarjetas Veroboard, que ya vienen de fábrica con pistas de cobre y paralelas e hileras de taladros distribuidos uniformemente, requieren un soldador de baja potencia, normalmente de 15 watios. Por la limitación de esta potencia, el calor que genera se expande con lentitud, y por tanto sirve mejor para los fines de soldaduras puntuales. No obstante, puede ser conveniente proveernos de un soldador de mayor potencia, de unos 30 a 45 watios, que nos facilitará la soldadura

bien

cuando existan grandes masas metálicas y el calor se disipe con rapidez.

Algo recomendable es comprarnos un modelo de puntas intercambiables, de tal manera que podamos utilizar la punta más adecuada para cada trabajo. Cuanto menor es el tamaño de la punta, menor es también el calor que almacena. Las puntas pequeñas, para soldaduras pequeñas y las grandes, para soldar conmutadores o terminales. Algunos soldadores de 30 watios disponen de un tipo de punta en forma de H, que sirve para desoldar de una sola vez las 14 ó 16 patillas de un circuito integrado DIL, en lugar de una a una, operación más trabajosa y arriesgada, por la cantidad de tiempo que se tiene caliente el *chip*.

Aspecto fundamental para la buena soldadura es también el estaño. En los comercios del ramo nos venderán un tipo de soldadura adecuada para fines electrónicos. Su composición no sólo es el estaño; lleva aleado plomo en proporciones centesimales 60/40. Para facilitar que el estaño fundido corra con mayor facilidad, lleva un alma de resina, que desprende un olor característico cuando funde. También protege a los terminales del componente



contra la oxidación durante el proceso de la soldadura. Cuando se enfría, la resina solidifica, tomando un aspecto pardusco, que aunque afea el acabado, se puede utilizar disolvente en cualquier droguería y, con ayuda de un pincel grueso, eliminarla de la placa.

El estaño se vende con distintos grosores; para los trabajos de microelectrónica es aconsejable uno de pequeño diámetro.

Después de conducir el trabajo, es importante limpiar la resina depositada en la punta del soldador antes de que se enfríe. Para ello se podría utilizar algún papel de tipo grueso o, mejor, un cepillo de cerdas metálicas, procurando utilizar la mínima energía al frontarlo.

Algunas bases a la venta para ser utilizadas como soporte para el soldador llevan una pequeña esponja destinada a esta tarea de limpieza. Personalmente no es mi gusto.

Los componentes electrónicos, normalmente soportan el calor generado durante un tiempo entre los 3 y los 10 segundos, sin sufrir daño. Si por cualquier causa fuera necesario más tiempo para asegurar la soldadura, es conveniente dejar que el calor se disipe durante un rato y volver a insistir posteriormente.

Muchos aficionados doblan los terminales que salen de los componentes con los dedos. Para obtener un acabado más atractivo es aconsejable tomar cada uno de los terminales con unas pinzas de punta plana, calcular la distancia del terminal al taladro del circuito impreso y, por ese punto, doblar el resto del terminal en 90°. Se doblará siempre por el extremo que queda libre, no por el lado del terminal.

Los componentes pueden, a veces, llevar una fina y transparente capa de barniz, grasa u óxido o cualquier otra sustancia. Por ello es conveniente limpiar los terminales; deberían limpiarse antes de la soldadura. Una goma de borrar para tinta se mostrará particularmente útil.

Haciéndole un corte perpendicular, como muestra la figura, se introducirá el terminal por él, tirando y empujando hasta asegurar que está limpio. Un terminal sucio puede producir un falso contacto. Muchas veces no es realmente necesaria esta limpieza, pero vale más prevenir.

Por su lado, conviene lavar con un jabón desengrasante la tarjeta de circuito impreso por el lado de las pistas.

Con un trozo de tarjeta Veroboard practicaremos la soldadura de distintos componentes, siguiendo las normas que veremos a continuación.

El misterio de soldar bien

Si trabaja en una mesa que pueda dañarse, no olvide protegerla con algunos periódicos o algo similar.

Conecte el soldador y deje que pase un tiempo prudencial para que la punta adquiera suficiente temperatura. Si intentamos utilizarlo apenas pueda derretir el estaño, las probabilidades de un falso contacto se multiplican. Unos cinco minutos suele ser suficiente.

En primer lugar, se doblarán los terminales del componente como describimos anteriormente. Una vez hecho, se inserta en los taladros correspondientes de la tarjeta de circuito impreso, doblándolos ligeramente, lo suficiente para que no se caigan al voltear la tarjeta.

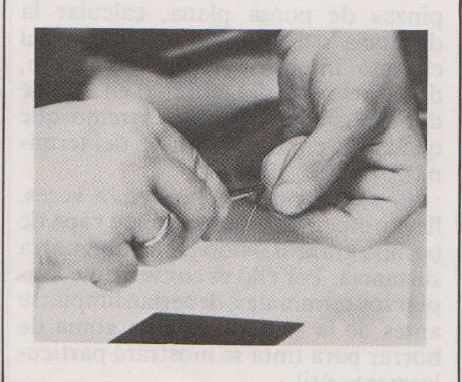
En el mercado existen algunos artefactos que facilitan el manejo y sujección del circuito integrado durante la soldadura, pero aparte de tener un precio algo elevado, no son especialmente relevantes sino se trabaja masivamente. Antes de utilizar el soldador, es necesario asegurarse de que la punta está correctamente plateada con estaño, sobre todo la parte que estará en contacto con la unión componente-pista de cobre. En un soldador nuevo esto se consigue mediante la aplicación de pequeñas cantidades de hilo de estaño cuando el soldador tiene una temperatura elevada, según ilustra la figura 3.

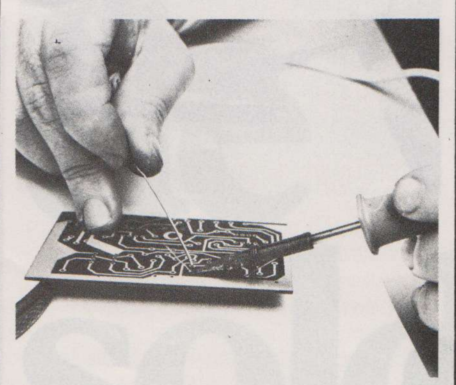
La punta del soldador se colocará de tal forma que toque tanto en la pista como en la parte del terminal más próxima a ella. Se deja que transmita su calor a ambos durante aproximadamente un par de segundos y, a continuación, se acerca el hilo de estaño (figura 4). Al fundirse, irá fluyendo como si de un líquido se tratase, hasta "mojar" ambas partes a soldar.

En el punto de unión, se formará una especie de montañita, cuya base se sitúa en la pista de cobre y va creciendo en torno al soldador (figura 5).

Podría ocurrir que la forma tomada por el estaño parezca una gota, con bordes redondeados. Esto se produce por dos causas. Bien la soldadura es correcta, pero hemos aplicado demasiada cantidad de estaño, y ese sobrante ha tomado dicha forma; o estamos ante una soldadura fría. Esto quiere decir que, si por ejemplo la pista no estaba lo suficientemente caliente antes de aplicar el estaño, cuando se ha puesto en contacto con ella ha absorbido el calor de la pista con intención de fundirse —el segundo principio de la termodinámica dice que el calor fluye de los cuerpos más calientes a los más fríos—, y se han acabado enfriando los dos, sin conseguir la perfecta unión.

El soldador seguirá aplicado durante algunos segundos más, hasta que veamos que el estaño quede en una posición más o menos estable. Una recomendación: no soplar nunca para





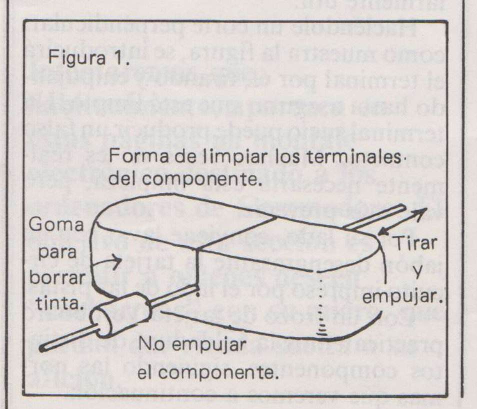


Figura 1.

Figura 2.

Doblar los terminales para mantener el componente en su sitio.

Cara del cobre.

Tarjeta del circuito impreso.

Zócalo para circuito integrado.

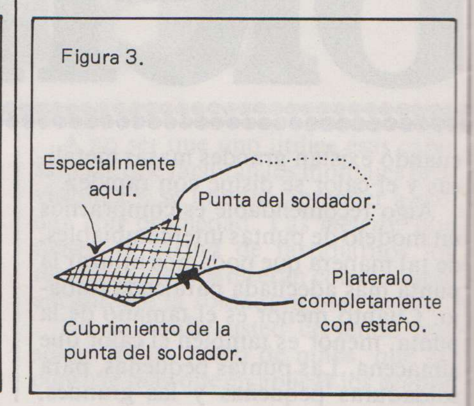


Figura 3.

que enfríe antes la unión, el riesgo de soldadura fría vuelve a aparecer.

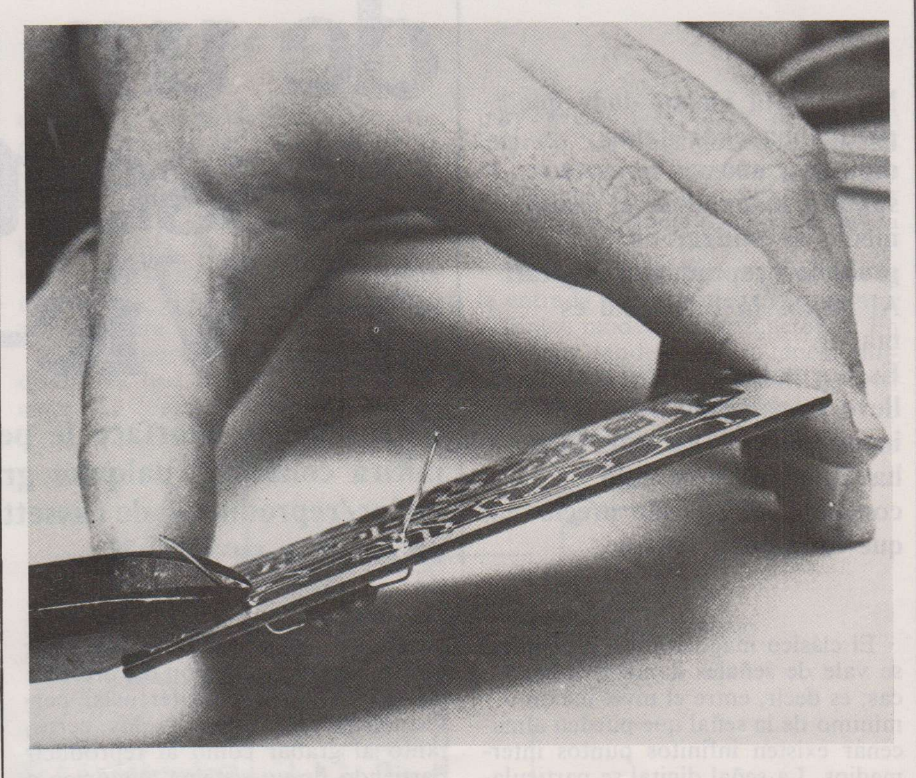
Varias veces hemos mencionado la "soldadura fría", pero nos gustaría saber cómo se detecta. Bueno, pues en primer lugar mediante la observación visual antes aludida. A continuación, tomando la parte del terminal que sobra, se tirará en sentido vertical hacia arriba. En muchos casos, si la soldadura es fría, termina cediendo la unión. Pero, ¡mucho cuidado! No tire demasiado fuerte, se podría llevar parte de la pista con la soldadura correspondiente.

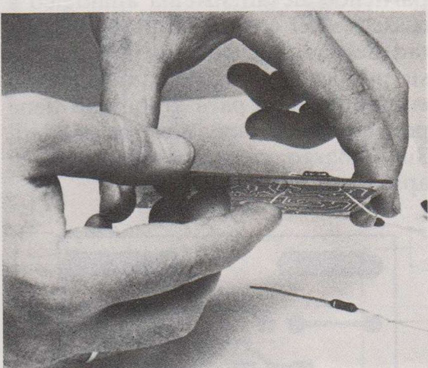
Una vez enfriada la unión, se podrán cortar los rabos sobrantes con un alicate de corte. No es aconsejable apurar demasiado.

Haga lo mismo con todos los componentes. Es recomendable seguir un orden durante la soldadura, dejando los componentes semiconductores, más propensos a sucumbir bajo la influencia del calor, para el último lugar.

Cuando hayamos soldado todos, la inspección ocular tomará el relevo. Se comprobará si cada componente ha sido colocado en su lugar correspondiente, si la disposición de las patillas de los componentes activos (transistores, diodos y circuitos integrados) es la que debiera.

Por último se puede limpiar toda la tarjeta, por el lado del cobre, con el disolvente y el pincel, como decíamos antes. Mucho cuidado, esta sustancia podría disolver determinadas partes de algunos componentes.





Una última revisión ocular siempre es recomendable. Esta vez conviene comprobar que las pistas no se tocan por efecto del estaño, etc.

Siguiendo estas observaciones, es prácticamente seguro que el montaje funcionará a la primera, siempre que los componentes no sean defectuosos.

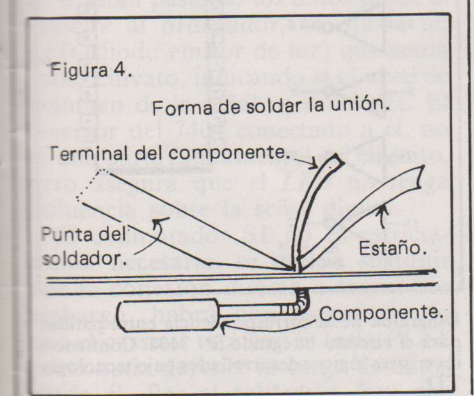


Figura 4.

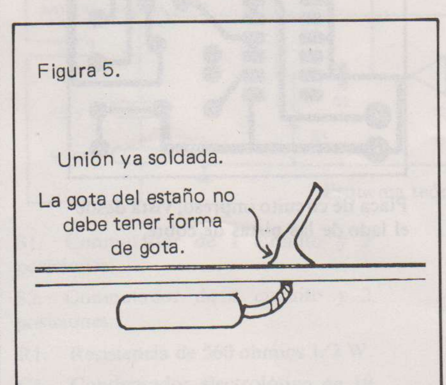


Figura 5.

Figura 6.

Conexión de un cassette al Vic-20

No cabe la menor duda que para almacenar datos digitales, unos y ceros, en soporte magnético, el mejor medio es utilizar un grabador/reproductor digital. Al menos la fiabilidad es mayor.
Es normal que toda ventaja lleve emparejado su inconveniente. En este caso hablamos de una viejo conocido, el elevado precio que la calidad impone.

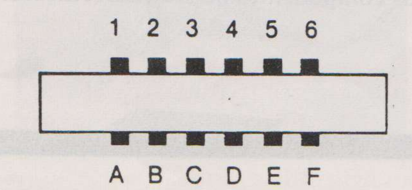
Este sencillo interface le permitirá conectar cualquier grabador/reproductor de cassettes estándar a su VIC-20.

El clásico magnetófono a *cassettes* se vale de señales llamadas analógicas; es decir, entre el nivel máximo y mínimo de la señal que pueden almacenar existen infinitos puntos intermedios. La señal digital se particulariza por su todo o nada, nivel máximo o nivel mínimo de señal. Los puntos intermedios no existen, y si aparecen puedne provocar la confusión. Es por esto que la fiabilidad de la grabación digital en un sistema analógico, incluso de muy alta calidad, puede conducir a errores. Lo cual no quiere decir que no sirvan. La misión del presente *interface* consiste en diferenciar perfectamente los unos de los ceros, tanto al grabar como al reproducir partiendo de un sistema analógico a cassettes. El costo y limitada dificultad del montaje nos han inclinado a su publicación. Anticipamos que la cinta empleada para grabar no tiene porqué ser de muy elevada calidad. Sin embargo, es recomendable que el aparato no lo sea de baja.

La salida del **VIC-20** fluctúa entre 0 y + 4 voltios y en cuanto a las señales que recibe, los niveles recomendados son 0 y — 4 voltios.

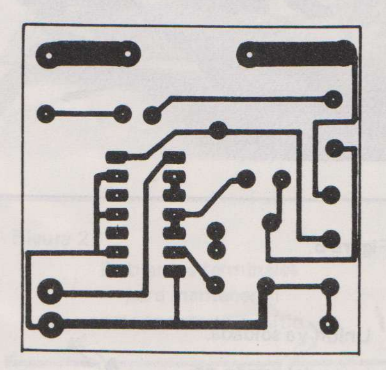
En el esquema teórico aparece a la izquierda el conector de seis terminales, que se conectará en la parte posterior del ordenador, observándose la ranura que asegura que se enchufará en posición correcta. A la derecha, las salidas que habrán de conectarse al cassette.

El montaje emplea solamente un



Patilla	VIC-20	Commodore 64
A y 1	Masa	Masa
B y 2	+ 5V.	+ 5V.
C y 3	Motor cassette	Motor cassette
D y 4	Lectura cassette	Lectura cassette
E y 5	Escritura cassette	Escritura cassette
F y 6	Commutador cassette	Conmutador cassette

Conexiones que ofrece el port destinado al cassette. Su configuración externa es la misma para el VIC-20 y el Commodore 64, según puede observarse en el cuadro anexo.



Placa de circuito impreso, vista desde el lado de las pistas de cobre.

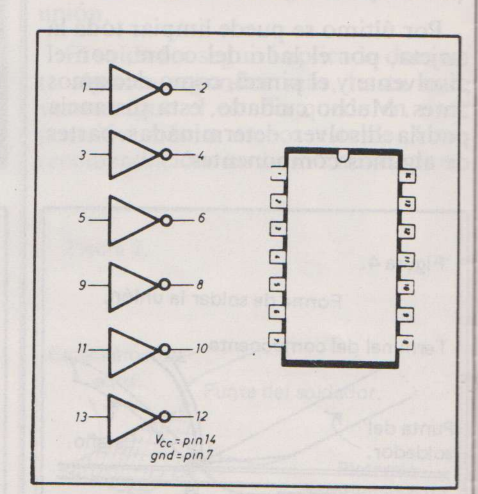


Diagrama de la correspondencia entre patillas para el circuito integrado SN 7404. Contiene 6 inversores lógicos desarrollados bajo tecnología TTL.

circuito integrado de 14 patillas. Contiene seis inversores lógicos, y su nomenclatura, para adquisición en tiendas especializadas, es 7404; pudiendo llevar delante otras letras, tales como SN, que indican el fabricante de procedencia. El chip que contiene está fabricado bajo la tecnología *TTL (Transistor-Transistor-Logic).*

El condensador C1 actua como separador eléctrico entre el ordenador y el magnetofón, sin embargo permite el paso de las señales digitales que, serán grabadas, por la entrada del micrófono externo. Para cargar datos, previamente grabados en la cinta, se recurre a la salida de auricular. Una vez que la señal ha pasado a través de dos inversores lógicos del c.i., ataca una red formada por el condensador electrolítico C2 y el diodo D1, consiguiendo los niveles de tensión 0 y —4 V.

Si el *cassette* dispone de entrada "remota", para accionar o detener el motor a distancia (muchos micrófonos suelen disponer en su carcasa de este interruptor), el **VIC** tendrá completo dominio sobre el momento de lectura de los datos. Para ello existe otra línea, que lleva intercalada un fusible, cuya misión consiste en limitar la corriente máxima de alimentación al motor, hasta un límite de aproximadamente 100 miliamperios.

Para efectuar la lectura, se recomienda que el control de volumen se sitúe en torno al número 8.

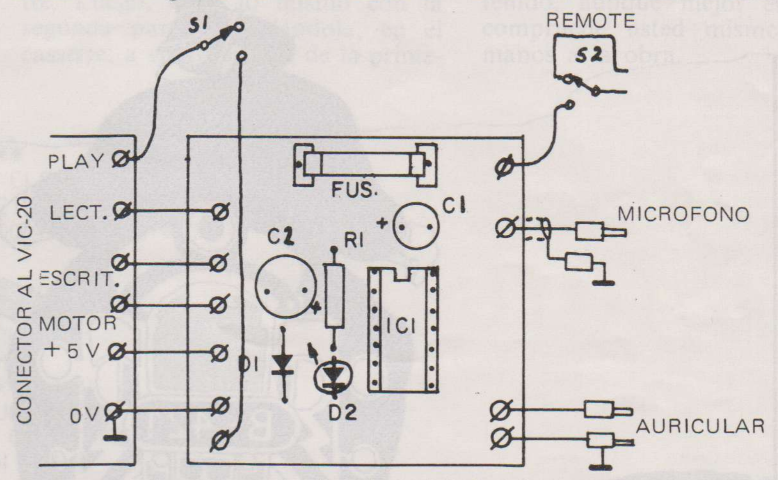
Para conocer en cualquier momento si están pasando los datos desde el *cassette* al ordenador, se utiliza un *LED* (diodo emisor de luz) que actua como chivato, indicando si el nivel de volumen de la señal es suficiente. El inversor del 7404 conectado a él, no es una parte fundamental del circuito, pero asegura que el *LED* no tenga influencia sobre la señal digital.

El conmutador S1 no es estrictamente necesario, se podría sustituir por la conexión directa a masa. Sin embargo, habrá que asegurarse de que el cassette está con la tecla *PLAY* presionada durante la carga de datos desde él. Por el contrario, hay que asegurarse que está presionado *RECORD* durante la grabación en la cinta, antes de presionar la tecla Return.

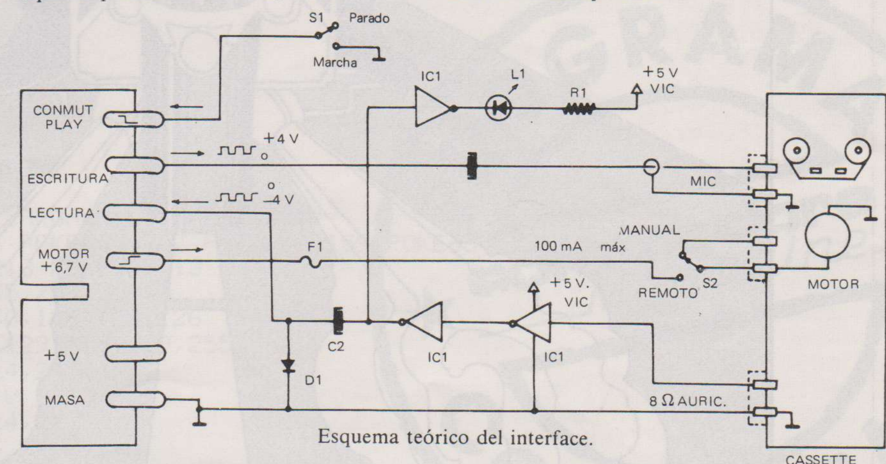
El interruptor S2 se utiliza al rebobinar la cinta. Es recomendable ponerlo en Remoto inmediatamente después de utilizarlo en la posición manual para el rebobinado.

En la entrada de micrófono se observará que aparece un cable coaxial. Su longitud será la necesaria para llegar a la placa de circuito impreso. Sin embargo, no se conectará a la masa del *interface,* puesto que podría aparecer ruido, producido por la realimentación a través de la salida de auricular.

El mando a distancia, por medio de la entrada *remote,* puede variar entre distintos modelos de magnetofón. En los casos donde aparezca mucha dificultad, será más aconsejable recurrir al control manual directo.



Esquema práctico del montaje, visto desde el lado de los componentes.



Esquema teórico del interface.

S1. Conmutador de 1 circuito y 2 posiciones.

S2. Conmutador de 1 circuito y 2 posiciones.

R1. Resistencia de 560 ohmios 1/2 W.

C1. Condensador electrolótico de 10 F, 16 V.

C2. Condensador electrolítico de 47 F, 16 V.

D1. Diodo de silicio 1N914.

L1. LED rojo.

F1. Fusible de 100 mA.

IC1. Circuito integrado TTL 7404.

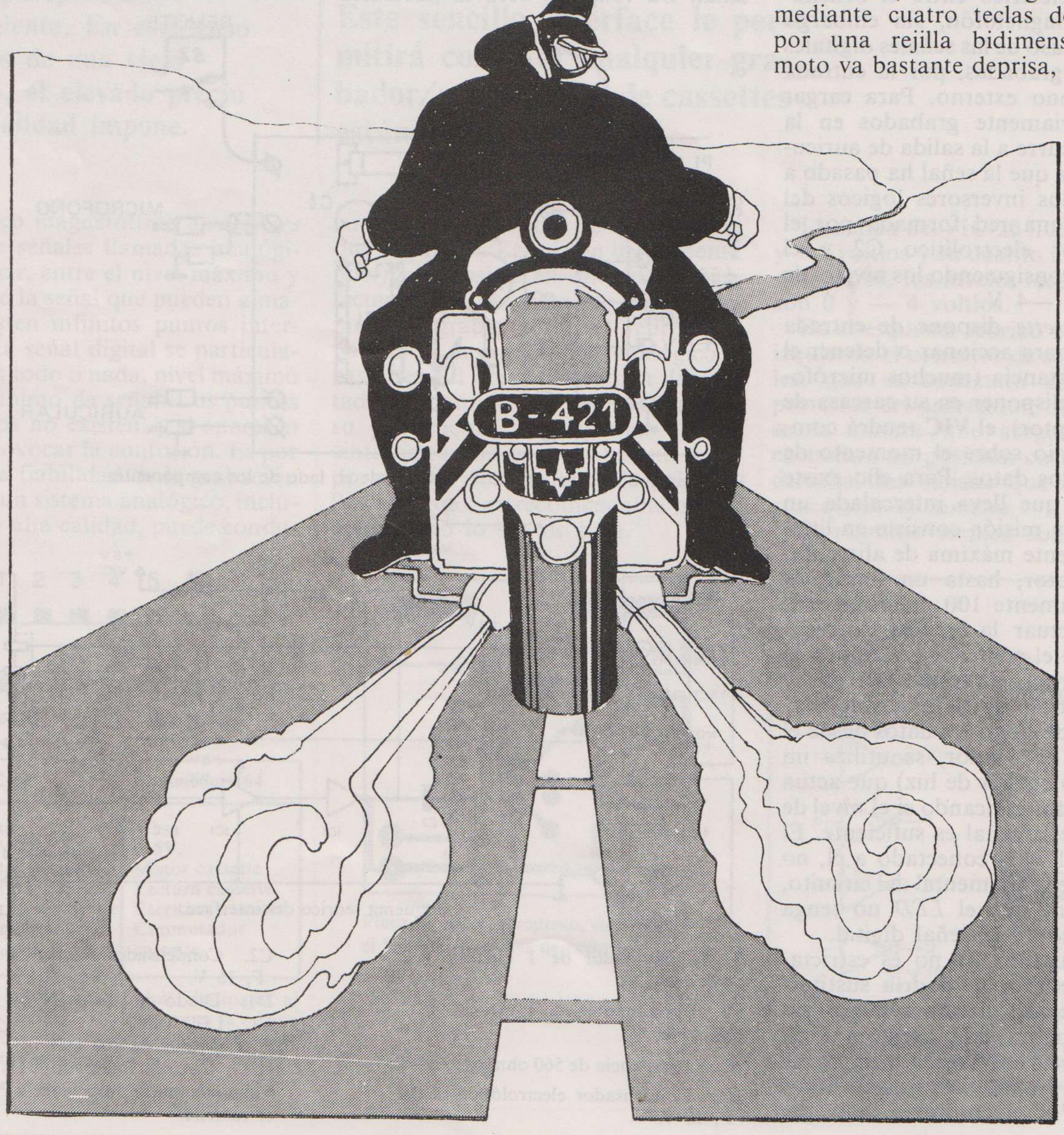
S1 simula que se pulsa la tecla PLAY en el cassette.

Juegos

Tron

Si usted vio la película "TRON" recordará sin duda las increíbles secuencias en las que, el protagonista, conduce una moto, a enorme velocidad, por una rejilla bidimensional, tomando curvas en ángulo recto, y dejando tras de sí, en su evolución por la rejilla, una pared sólida contra la que se estrellan sus perseguidores.

Con este programa usted puede emular las hazañas del protagonista. En esencia, usted dirige una moto, mediante cuatro teclas del teclado, por una rejilla bidimensional. La moto va bastante deprisa, por lo que



son necesarios unos buenos reflejos para conseguir llevarla por donde uno quiere, evitando estrellarse. En algunas de las células de la rejilla hay hombrecitos esperándole, que usted deberá recoger. El problema es que, tanto los bordes de la pantalla, como la traza que deja la moto, son sólidos muros y si choca con ellos le destruirán, con lo que termina el juego. No es raro, al principio, que en sus evoluciones por la rejilla, quede usted encerrado en su propia traza y acabe estrellándose sin remedio. Cada vez que recoja a todos los hombrecitos de una rejilla, pasará a otra nueva, con un hombrecito más. A medida que coja práctica, puede que quiera probar con la versión difícil. En este caso, no sólo tendrá que evitar los límites de la rejilla y su propia traza, sino que tendrá que vérselas con una serie de sólidos bloques que aparecerán, diabólicamente distribuidos, para que usted se estrelle. El programa viene dividido en dos partes. Teclee la primera parte y guárdela en el cassette. Luego, haga lo mismo con la segunda parte, guardándola, en el cassette, a continuación de la primera. Para jugar, cargue la primera parte en su ordenador y escriba RUN; aparecerán las instrucciones del juego y el programa le dirá que ponga en marcha el cassette. Hágalo, con lo que se cargará la segunda parte y dará comienzo el juego. Si para el programa podrá volver a correrlo escribiendo RUN pero siempre que lo cargue de la cinta, deberá hacerlo siguiendo los pasos indicados.

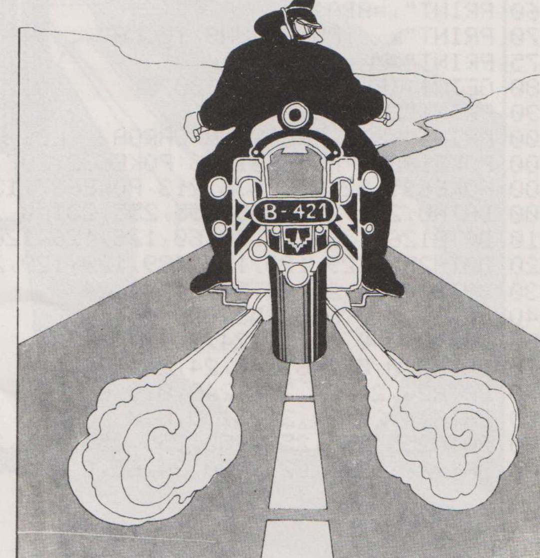
El juego es verdaderamente entretenido, aunque mejor está que lo compruebe usted mismo, así que, manos a la obra.




```
900 REM***PRIMERA PARTE***
1000 POKE52,28:POKE56,28:CLR
1010 FORI=7168TO7256:POKEI,PEEK(I+25600):NEXT
1020 FORJ=7168TO7256:READQ:POKEJ,Q:NEXT
5000 PRINT"]■"
5010 POKE36879,8
5020 PRINT"    ■TRON!!"
5030 PRINT"■CONDUCES UNA MOTO"
5040 PRINT"POR LA REJILLA"
5050 PRINT"■DEBES RECOGER A LA"
5060 PRINT"GENTE QUE ENCUENTRES"
5070 PRINT"■LA MOTO DEJA UNA TRAZA"
5080 PRINT"]SI CHOCAS CON ELLA"
5090 PRINT"TE DESTRUIRA"
5130 PRINT"Z=IZQUIERDA"
5140 PRINT"X=DERECHA"
5150 PRINT"L=ARRIBA"
5160 PRINT",=ABAJO"
5170 PRINT"■   PULSA UNA TECLA"
5175 PRINT"■■   "
5180 GETA$:IFA$=""THEN5180
5190 PRINT"]"
5200 PRINT"ESTE PROGRAMA CARGA   LOS GRAFICOS"
5300 POKE198,7:POKE631,76:POKE632,207:POKE633,159:POKE634,13
5400 POKE635,82:POKE636,213:POKE637,13
9000 DATA0,231,255,255,255,255,231,0
9010 DATA126,126,126,60,60,126,126,126
9020 DATA255,129,129,129,129,129,129,255
9030 DATA24,24,24,24,24,24,24,24
9040 DATA0,0,0,255,255,0,0,0
9050 DATA0,0,0,31,31,24,24,24
9060 DATA0,0,0,248,248,24,24,24
9070 DATA24,24,24,248,248,0,0,0
9080 DATA24,24,24,31,31,0,0,0
9090 DATA255,255,255,255,255,255,255,255
9100 DATA28,28,8,62,8,20,34,65,0
```

Juegos

```
0 QWE=RND(1-TI)
1 CLR
2 PRINT"[CLR][RVS]FACIL ([RVS]1[OFF])O DIFICIL ([RVS]2[OFF])":INPUT TYU
3 IF TYU<1ORTYU>2 THEN2
4 REM***SEGUNDA PARTE***
5 PRINT"[CLR]":POKE36878,15:POKE36869,255
10 A=8174:GRID=1:CH=1:DF=1:WT=3:NW=3:D=-22:SC=0:MAN=1
45 POKE36879,56
50 FORN=7680TO8185:POKEN,2:NEXT
80 FORN=38400TO38905:POKEN,6:NEXT
96 FORN=7680TO8164STEP22:POKEN,9:NEXT
97 FORN=38400TO38884STEP22:POKEN,0:NEXT
98 FORN=1TOMAN:RP=INT(RND(1)*506)+7680:IFPEEK(RP)<>2 THENRP=RP+1
99 POKE RP,10:NEXTN
100 IF TYU=1 THEN 103
101 FORN=1TO10:SP=INT(RND(1)*506)+7680:IFPEEK(SP)<>2THENSP=SP+1
102 POKESP,230:POKESP+30720,0:NEXT
103 POKEA,CH:POKEA+30720,2:POKEA+(-D),WT
104 POKE36874,196:POKE36875,196:POKE36876,176
105 OD=D
106 IFNW=0THEN110
107 WT=NW
110 GETA$
120 IFA$="Z"THEN D=-1:WT=4:CH=0
130 IFA$="X"THEN D=1:WT=4:CH=0
140 IFA$="L"THEN D=-22:WT=3:CH=1
150 IFA$=","THEN D=22:WT=3:CH=1
160 A=A+D
162 IFPEEK(A)>2ANDPEEK(A)<10 ORPEEK(A)=230THEN4000
163 IFPEEK(A)=10THENGOTO6100
165 GOTO3000
180 IFA<7680 ORA>8185THENGOTO4000
190 GOTO103
3000 IFOD=-22ANDD=-1THENNW=4:WT=6
3010 IFOD=22ANDD=-1THENNW=4:WT=7
3020 IFOD=-22 ANDD=1THENNW=4:WT=5
3030 IFOD=22ANDD=1THENNW=4:WT=8
3040 IFOD=1ANDD=22THENNW=3:WT=6
3050 IFOD=1ANDD=-22THENNW=3:WT=7
3060 IFOD=-1ANDD=-22THENNW=3:WT=8
3070 IFOD=-1ANDD=22THENNW=3:WT=5
3080 GOTO180
4000 FORKN=1TO10
4002 QW=128
4010 FORN=0TO7
4020 POKE(A+-D)+30720,N
4030 POKE36877,QW
4035 QW=QW+5
4040 NEXT
4050 NEXT
4055 POKE36877,0
4060 FORMJ=1TO250:NEXT
4070 POKE36869,240
4080 PRINT"[CLR][RVS]"
4090 POKE36879,8
```



```
4100 PRINT"   TRON!!"
4110 PRINT"TU PUNTUACION=";SC
4120 IFSC>HSTHENHS=SC
4130 PRINT"RECORD=";HS
4140 PRINT"OTRO JUEGO (S/N)"
4150 GETA$:IFA$=""THEN4150
4151 RESTORE
4160 IFA$="S"THENGOTO4200
4170 IFA$="N"THENPOKE36874,0:POKE36875,0:POKE36876,0:POKE36879,27:PRINT"ADIOS!
":END
4180 GOTO4150
4200 PRINT"FACIL1 ODIFICIL2 ?"
4210 INPUT TYU
4215 IFTYU<1 OR TYU>2THEN4210
4220 PRINT"":POKE36869,255:GOTO5
6100 DF=DF+1
6102 SC=SC+10
6105 IFDF<MANTHEN165
6110 PRINT"":POKE36874,0:POKE36875,0:POKE36876,0
6120 PRINT"REJILLA";GRID;"LIMPIA"
6125 FORRT=1TO2500:NEXT
6130 DF=0:MAN=MAN+1:GRID=GRID+1:SC=SC+100:A=8174:D=-22:WT=3:CH=1:NW=3:GOTO45
```

Las torres de Brahma

Se supone que Brahma, la suprema deidad hindú, propuso a sus discípulos un interesante pasatiempo, probablemente para que llenaran sus largas horas de meditación.

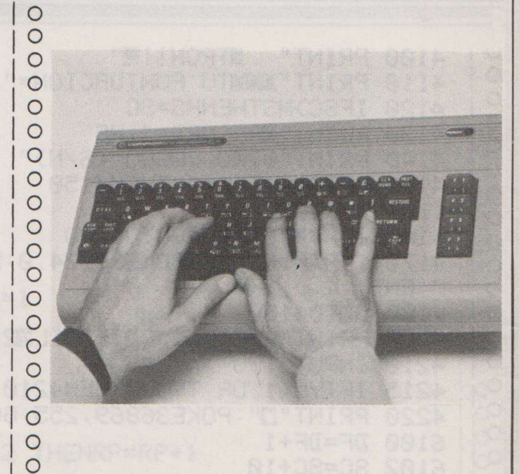
El problema, que puede intentar resolver en su Commodore 64 con este programa, consiste en transferir una serie de anillos de diámetros decrecientes, de una varilla A a otra varilla C, con la ayuda de una varilla intermedia B. Pero eso sí, sólo moviendo un anillo por vez, y con la condición de no situar nunca, en ninguna de las varillas, un anillo más ancho encima de otro más estrecho. El programa le recordará estas instrucciones y le dirá cómo visualizar el desarrollo del juego. Intente resolver el problema. Si lo consigue, el programa le dirá cuántos movimientos ha empleado y cuánto tiempo le ha llevado encontrar la solución.

Parece ser que los discípulos de Brahma, pasaron muy buenos ratos buscando la solución de este problema, incluso, descuidando a veces sus obligaciones sacerdotales.

```
100 PRINTCHR$(147);"LA TORRE DE BRAHMA"
110 DIMA(3,9)
120 PRINT
130 PRINT"ES COMO ESTA(CON 5 ANILLOS):-"
140 PRINT
150 PRINT"         A         B        C"
160 PRINT"        1A1        B        C"
170 PRINT"       22A22       B        C"
180 PRINT"      333A333      B        C"
190 PRINT"     4444A4444     B        C"
200 PRINT"    55555A55555    B        C"
210 PRINT"********************************"
220 PRINT
230 PRINT"EL OBJETO DEL JUEGO ES TRASPASAR"
240 PRINT"LOS ANILLOS DE LA COLUMNA 'A' A LA 'C',"
250 PRINT"DE UNO EN UNO CADA VEZ."
260 PRINT
270 PRINT"EN NINGUN MOMENTO PUEDE HABER UN"
280 PRINT"ANILLO GRANDE ENCIMA DE UNO MENOR."
290 PRINT
300 PRINT"PUEDES VER COMO VAS EN CUALQUIER"
310 PRINT"MOMENTO,PULSANDO 'V' COMO RESPUESTA"
311 PRINT"A LA PETICION 'DESDE'."
320 PRINT
330 PRINT
340 INPUT "CUANTOS ANILLOS QUIERES (5-9)
    *";Z$
350 IF Z$="*"THENPRINT"":GOTO340
360 Z=VAL(Z$)
370 IFZ<5 THEN PRINT"":GOTO340
```

Juegos

```
380 IFZ>9THENPRINT"TJ":GOTO340
390 PRINTCHR$(147)
400 TI$="000000"
410 FORX=1TO3
420 FORY=1TOZ
430 IFX=1THENA(X,Y)=Y
440 IFX>1THENA(X,Y)=0
450 NEXTY
460 NEXTX
470 INPUT "MOVER UN ANILLO DESDE ▶▶*■■■";J$
480 Q=Q+1
490 J$=LEFT$(J$,1)
500 IFJ$="*"THENPRINT"TJ":GOTO470
510 IFJ$="V"THENPRINTCHR$(147):GOTO760
520 IFJ$="S"THEN990
530 IFJ$<"A"THEN470
540 IFJ$>"C"THEN470
550 V=ASC(J$)-64
560 FORY=1TOZ
570 IFA(V,Y)>0THENT=A(V,Y):A(V,Y)=0:GOTO610
580 NEXTY
590 PRINT"NO HAY ANILLOS EN ESA COLUMNA"
600 GOTO470
610 INPUT"HASTA  ▶▶*■■■";K$
620 K$=LEFT$(K$,1)
630 IF K$="*" THEN PRINT"TJ":GOTO610
640 IFK$="S"THEN990
650 IFK$<"A"THEN610
660 IFK$>"C"THEN610
670 U=ASC(K$)-64
680 FORW=ZTO1STEP-1
690 IFA(U,W)=0THENIFW=ZTHENA(U,W)=T:GOTO470
700 IFA(U,W)=0THENIFW<ZTHENIFA(U,W+1)<T THEN730
710 IFA(U,W)=0THENA(U,W)=T:GOTO840
720 NEXTW
730 PRINT"EL ANILLO EN ESA COLUMNA ES MENOR"
740 A(V,Y)=T
750 GOTO470
760 FORY=1TOZ
770 FORX=1TO3
780 PRINTTAB(5*X);A(X,Y);
790 NEXT X
800 PRINT
810 NEXTY
820 PRINT
830 GOTO470
840 FORY=1TOZ
850 IFA(3,Y)<>YTHEN470
860 NEXTY
870 PRINTCHR$(147)
880 PRINT"ENHORABUENA!,LO HAS CONSEGUIDO"
890 PRINT"Y SOLO HAS TARDADO ";
900 PRINTMID$(TI$,3,2);"'";RIGHT$(TI$,2);"''"
910 PRINT"Y ";Q;" MOVIMIENTOS"
920 PRINT
930 INPUT "OTRA VEZ ▶▶*■■■";A$
940 IFLEFT$(A$,1)="*" THEN PRINT"TJ":GOTO930
950 IF LEFT$(A$,1)="S" THEN330
960 IF LEFT$(A$,1)<>"N" THEN930
970 PRINT"HASTA LUEGO..."
980 END
990 PRINT"LO SIENTO,SO FAR IT"
1000 GOTO890
```






ZX
La nueva revista para usuarios del ZX-81 y SPECTRUM
Programas/Juegos/Montajes/Còdigo Màquinà
Año I / No. 3 - 200 Ptas.
ZX
REVISTA PARA LOS USUARIOS DE ORDENADORES SINCLAIR
¡Ya está a la venta!
Cómprela en su quiosco o solicítela a:
ZX
Jerez, 3
Tel: 91-457 45 66
Madrid 16
POR DENTRO
Tamaño real: 19,5 x 26,

El mejor ordenador personal del Mundo sólo cuesta 79.900 ptas.

EL COMMODORE 64

1. Capacidad total de memoria RAM de 64 K. Interpretador BASIC extendido y sistema operativo residentes en ROM.

2. Dotado del más potente chip sintetizador de sonido diseñado hasta hoy, el COMMODORE 64 ofrece 3 voces totalmente independientes con una gama de 9 octavas. El programa puede controlar la envolvente, la afinación y la forma de onda de cada voz, convirtiendo al COMMODORE 64 en el mejor simulador de instrumentos.

3. Conectable directamente a toda una gama de periféricos, incluyendo unidad de discos, impresora de matriz de puntos o de margarita, plotter, comunicaciones locales y remotas..., y mucho más.

4. Pantalla de alta resolución en color con 320 × 200 puntos directamente direccionables. Capacidad en modo carácter de 25 líneas por 40 columnas.

5. El chip de video, único en su género, permite el uso de 8 «Sprites» (figuras móviles en alta resolución y color). Los «Sprites» pueden moverse independiente-mente por programa de «pixel» en «pixel».

6. Teclado profesional con mayúsculas y minúsculas, más 62 caracteres gráficos, todos ellos disponibles en el teclado y visualizables en 16 colores, en forma normal o bien en video invertido.

7. Encontrará a su disposición una completa gama de programas profesionales, incluyendo proceso de textos, sistemas de información, modelos financieros, contabilidad y muchas más aplicaciones.

8. Están en fase de desarrollo otros lenguajes tales como LOGO, COMAL, PILOT, etc.

9. Opción de un segundo procesador Z-80 para trabajar con sistema operativo CP/M (R).



MICROELECTRONICA Y CONTROL
c/ Taquígrafo Serra, 7, 5° Barcelona-29
c/ Princesa, 47, 3°, G Madrid-8

SLOGAN